

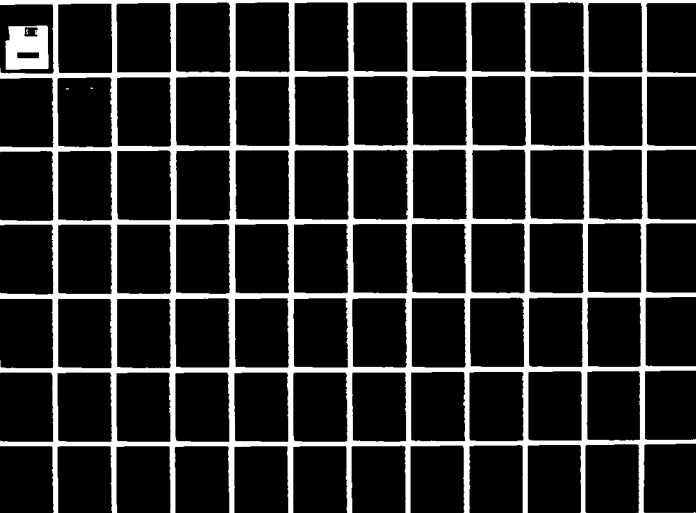
AD-A195 836

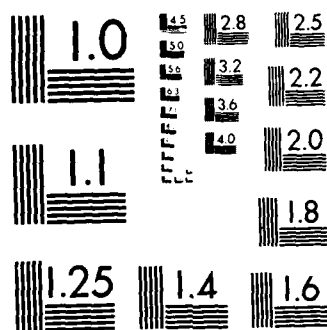
INTEGRATING INFORMATION SYSTEMS IN A MAJOR
DECENTRALIZED INTERNATIONAL OR. (U) MASSACHUSETTS INST
OF TECH CAMBRIDGE A GUPTA ET AL. DEC 87 HIT-KBIISE-7
DTR557-85-C-00083 F/G 12/7

1/3

UNCLASSIFIED

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

DTIC FILE COPY



Massachusetts
Institute of
Technology

Knowledge-Based
Integrated Information
Systems Engineering
(KBIISE) Project

Volume 7

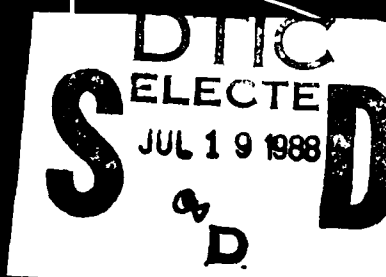
Integrating Information Systems in a Major Decentralized International Organization

(2)

Amar Gupta
Stuart Madnick

SERIES EDITORS

AD-A195 856



DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER KBIISE-7	2. GOVT ACCESSION NO. A195-532	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Integrating Information Systems in a Major Decentralized International Orgniazation		5. TYPE OF REPORT & PERIOD COVERED Part of Final Report; Sept 88 to Jan 88
7. AUTHOR(s) Amar Gutna and Stuart Madnick (editors)		6. PERFORMING ORG. REPORT NUMBER KBIISE-7
9. PERFORMING ORGANIZATION NAME AND ADDRESS Massachusetts Institute of Technology Cambridge, MA 02139		8. CONTRACT OR GRANT NUMBER(s) DTRS57-85-C-00083
11. CONTROLLING OFFICE NAME AND ADDRESS Transportation Systems Center Broadway Street Cambridge, MA 02142 (in conj. with USAF)		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Transportation Systems Center Broadway Street Cambridge, MA 02142		12. REPORT DATE December, 1987
		13. NUMBER OF PAGES 283
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE na
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES This volume is one of eight volumes prepared by MIT for the Department of Transportation and Department of Defense (US Air Force).		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Integration, knowledge, databases, systems engineering, methodologies, information modeling, heterogeneous database systems.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This volume examines the problems involved in integrating information systems currently in use in a large decentralized international organization. It is divided into three parts. The first part, "A Conceptual Model for Integrated Autonomous Processing," highlights the fact that the technical strategy was modulated by three key conflicting organizational forces -- autonomy, integration and evolution. A conceptual system architecture has been developed to provide a technological message-orientated and		

data-orientated response to meet the organizational needs.

The second part, "Gaining Strategic Advantage Through Composite Information Systems," identifies critical success factors for the successful deployment of Composite Information Systems (CIS) ideas and techniques. For this environment, a data-driven strategy offers the best chances for successful development of a modular and flexible infrastructure suited to the decentralized and highly autonomous structure of the organization.

The third part, "Integrating Systems for Financial Institutes Services Using Composite Information Systems," analyzes the three key organizational forces in greater detail. Autonomy, for example, is examined in terms of hardware control, operational control, transaction control, software control, data control, and management control. The forces of integration and evolution are also decomposed in a similar vein. The case study reveals that the operations of the organization can be improved by more foresight in certain areas.

Account No.	
NTIS	✓
DTIC	✓
Unrec'd	✓
Justified	
By	
Date	
Approved	
Date	
A-1	

INTEGRATING INFORMATION SYSTEMS IN A MAJOR DECENTRALIZED INTERNATIONAL ORGANIZATION

About This Volume

This volume examines the problems involved in integrating information systems currently in use in a large decentralized international organization. It is divided into three parts. The first part, "A Conceptual Model for Integrated Autonomous Processing," highlights the fact that the technical strategy was modulated by three key conflicting organizational forces – autonomy, integration, and evolution. A conceptual system architecture has been developed to provide a technological message-oriented and data-oriented response to meet the organizational needs.)

The second part, "Gaining Strategic Advantage Through Composite Information Systems," identifies critical success factors for the successful deployment of Composite Information Systems (CIS) ideas and techniques. For this environment, a data-driven strategy offers the best chances for successful development of a modular and flexible infrastructure suited to the decentralized and highly autonomous structure of the organization.

The third part, "Integrating Systems for Financial Institutions Services Using Composite Information Systems," analyzes the three key organizational forces in greater detail. Autonomy, for example, is examined in terms of hardware control, operational control, transaction control, software control, data control, and management control. The forces of integration and evolution are also decomposed in a similar vein. The case study reveals that the operations of the organization can be improved by more foresight in certain areas. Keywords: Knowledge Based

Integrated Information Systems Engineering (KBISE). (KR)

Table of Contents

	Page
SERIES EDITORS' NOTE	1
A CONCEPTUAL MODEL FOR INTEGRATING AUTONOMOUS PROCESSING (Technical Report #6)	5
GAINING STRATEGIC ADVANTAGE THROUGH COMPOSITE INFORMATION SYSTEMS (Technical Report #12)	41
INTEGRATING SYSTEMS FOR FINANCIAL INSTITUTIONS SERVICES USING COMPOSITE INFORMATION SYSTEMS (Technical Report #14)	147

SERIES EDITORS' NOTE

This book is one of eight volumes published by MIT as part of the Knowledge-Based Integrated Information Systems Engineering Project (KBIISE). In order to appreciate the papers in this book, it is necessary to be aware about the theme of the KBIISE project, its major objectives, and the different documents that summarize the research accomplishments to date.

Goal

The primary goal of the KBIISE project is to integrate islands of disparate information systems that characterize virtually all large organizations. The number and the size of these islands has grown over years and decades as organizations have invested in an increasing number of computer systems to support their growing reliance on computerized data. This has made the problem of integration more pronounced, complex, and challenging.

The need for multiple systems in large organizations is dictated by a combination of technical reasons (such as the desired level of processing power and the amount of storage space), organizational reasons (such as each department obtaining its own computer based on its function), and strategic reasons (such as the level of reliability, connectivity, and backup capabilities). Further, underlying trends in the information technology area have led to a situation where most organizations now depend on a portfolio of information processing machines, ranging from mainframes to minicomputers and from general purpose workstations to sophisticated CAD/CAM systems, to support their computational requirements. The tremendous diversity and the large size of the different systems make it difficult to integrate these systems.

Key Participants

The above problem is becoming increasingly evident in all large government agencies and in large development programs. In the fall of 1986, the U.S. Air Force (USAF) and the Transportation Systems Center (TSC) of the U.S. Department of Transportation approached M.I.T. to conduct and to coordinate research activity in this area in order "to develop the framework for a comprehensive methodology for large scale distributed, heterogeneous information systems which will provide: (i) the necessary structure and standards for an evolving top down global framework; (ii) simultaneous bottom up systems development; and (iii) migratory paths for existing systems."

Both USAF and TSC provided sustained assistance to members of our research team. In addition, Citibank and IBM provided some funds for research in very specific areas. One advantage of our corporate links was the opportunity to analyze and to generate case studies of actual decentralized organizational environments.

The research sponsors and MIT agreed that in order to deal with the heterogeneity issue in a meaningful way, it was important that a critical mass of influential individuals participate in the development of solutions. Only through widespread discussion and acceptance of a proposed strategy would it become feasible to deal with the major problems. For these reasons, a Technical Advisory Panel (TAP) was constituted. Nominees to the TAP included experts from academic and research organizations, government agencies, computer companies, and other corporations. In addition, several subcontractors, the primary one being Texas A&M University, provided assistance in specific areas.

Technical Outputs

The scope of the work included (i) technical issues; (ii) organizational issues; and (iii) strategic issues. On the basis of exploratory research efforts in all these areas, 24 technical reports were prepared. Eighteen of these reports were generated by MIT research personnel, and their respective areas of investigation are summarized in the figure on the opposite page.

The five technical reports, not represented in the figure, are as follows:

- #1. Summary.
- #2. Record of discussions held at the first meeting of the Technical Advisory Panel (TAP) on February 17, 1987.
- #3. Consolidated report submitted by Texas A&M University.
- #21. Annotated Bibliography.
- #23. Record of discussions held at the second meeting of the Technical Advisory Panel (TAP) on May 21 and 22, 1987.
- #24. Contributions received from members of the TAP highlighting their views on various aspects of the problem.

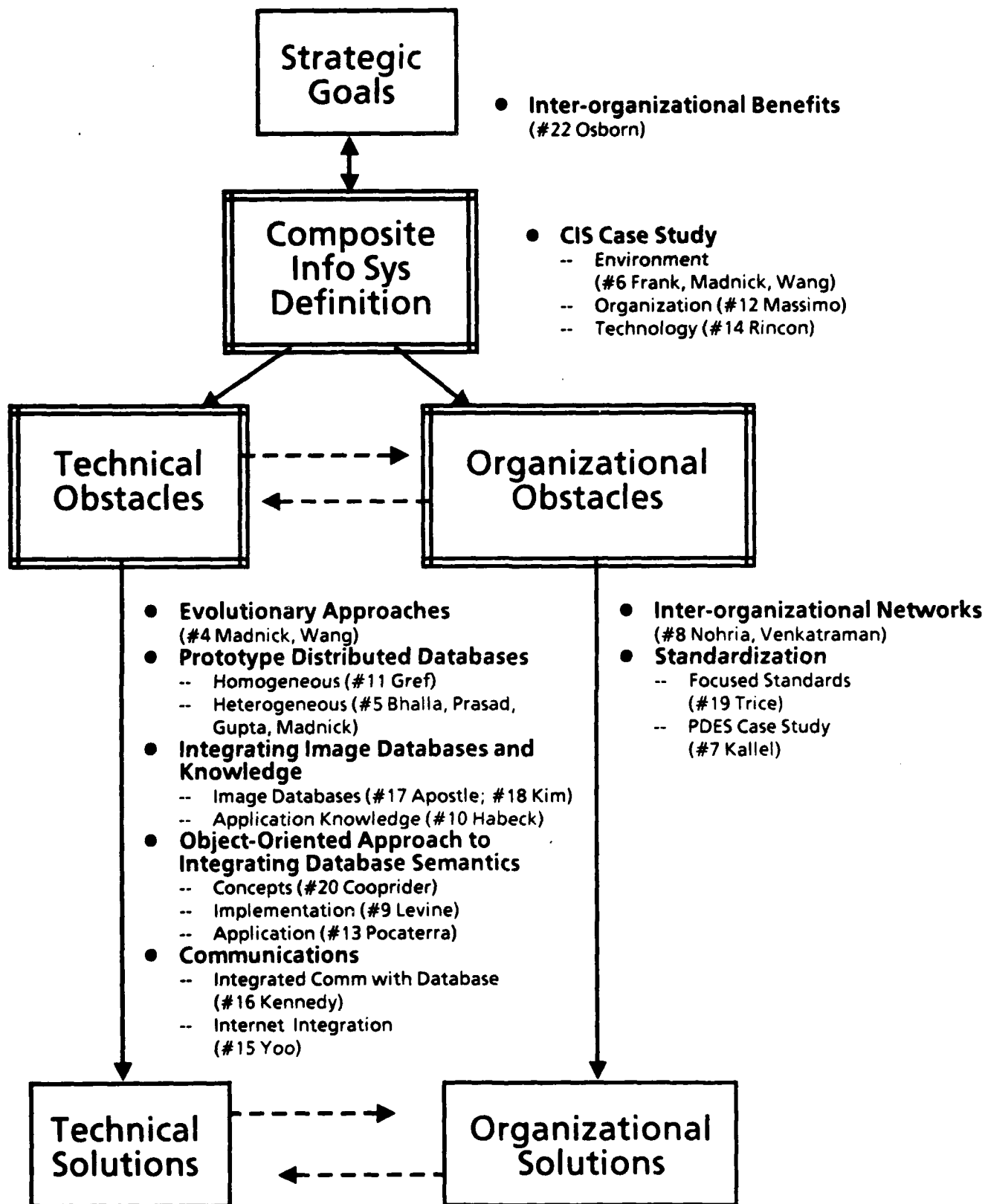
All the 24 technical reports have been edited and reorganized as an eight-volume set. The titles of the different volumes are as under:

1. KNOWLEDGE-BASED INTEGRATED INFORMATION SYSTEMS ENGINEERING-HIGHLIGHTS AND BIBLIOGRAPHY
2. KNOWLEDGE-BASED INTEGRATED INFORMATION SYSTEMS DEVELOPMENT METHODOLOGIES PLAN
3. INTEGRATING DISTRIBUTED HOMOGENEOUS AND HETEROGENEOUS DATABASES - PROTOTYPES
4. OBJECT-ORIENTED APPROACH TO INTEGRATING DATABASE SEMANTICS
5. INTEGRATING IMAGES, APPLICATIONS, AND COMMUNICATIONS NETWORKS
6. STRATEGIC, ORGANIZATIONAL, AND STANDARDIZATION ASPECTS OF INTEGRATED INFORMATION SYSTEMS
7. INTEGRATING INFORMATION SYSTEMS IN A MAJOR DECENTRALIZED INTERNATIONAL ORGANIZATION
8. TECHNICAL OPINIONS REGARDING KNOWLEDGE-BASED INTEGRATED INFORMATION SYSTEMS ENGINEERING

Volume 2 contains the report submitted by Texas A&M and Volume 8 highlights the views of members of the TAP. Activities described in the other 6 volumes have been conducted at MIT.

EXPLORATORY RESEARCH EFFORTS

3.



Acknowledgments

Funds for this project have been provided by U.S. Air Force, U.S. Department of Transportation (Contract Number DTRS57-85-C-00083), IBM, and Citibank. We thank all these organizations and their representatives for their support. In particular, we are indebted to Major Paul Condit of U.S. Air Force for his initiative in sponsoring this project, to Dr. Frank Hassler, Bud Giangrande, and Bob Berk of the Transportation Systems Center (TSC) for their support and assistance, to Professor Joseph Sussman, Director, Center for Transportation Studies (CTS) at MIT for his help and encouragement, and to all the individuals whose results have been published in this book.

We would welcome receiving feedback from readers of this book.

Amar Gupta and S.E. Madnick
Massachusetts Institute of Technology
Cambridge, Massachusetts.

A CONCEPTUAL MODEL FOR INTEGRATED AUTONOMOUS PROCESSING

WILLIAM FRANK, STUART MADNICK, AND Y. RICHARD WANG

This research effort is the first in a series that explores the development of Composite Information Systems (CIS) in a technologically and organizationally complex environment. Since access to the details of military applications was not possible due to time and security limitations, the particular situation studied is that of a major international bank where competitive pressure has encouraged the development and deployment of information technology.

Three key conflicting organizational forces were found to significantly impact the technical strategies:

1. Autonomy - is critical to each individual group to maximize flexibility and minimize risk since each manager is held completely accountable for the performance of his or her applications.
2. Integration - is important to the organization as a whole due to the needs to share information between applications.
3. Evolution - is necessary to respond to a rapidly changing marketplace and the realization that as applications become more sophisticated, there is more need to acquire information from other systems.

A conceptual system architecture was developed to provide a technological message-oriented and data-oriented response to these organizational needs by integrating only those elements critical to integration and allowing complete autonomy for the vast amount of the application specification, development, and operation. The functional components are separated into five layers, with the four integrating application-independent layers (*external interface, message control, data control, and shared-data resources*) surrounding the application processing components.

This system architecture was used in the development of two complex integrated application groups (i.e., each application group represented the integration of what were previously multiple independent applications). One application group was a high-volume transaction ensemble involving 20 gigabytes of data, the other application group was a sophisticated management reporting system involving over 1.5 gigabytes of data.

Although the system architecture was very effective at helping to integrate the components of each application group, the attempt to integrate the two application groups together was only partially successful at that time due to the autonomy culture. The system architecture is designed to be an evolutionary blueprint since the movement from complete autonomy to complete integration is slow and may not even be desirable. The approach of separating the external interfaces, message control, data control, and database components from the application processing does provide for high degrees of integration while preserving significant autonomy -- and the ability to evolve further in either direction.

TECHNICAL REPORT #6

A Conceptual Model for Integrated Autonomous Processing: An International Bank's Experience with Large Databases

1. Introduction

The successful exploitation of information technology requires a delicate and coordinated interplay of strategic application identification, organizational constraints and adjustments, and selection of appropriate technologies, as depicted in Figure 1. To understand these issues better a Strategic Applications, Technology, and Organization Research Initiative (Project SATORI) has been undertaken within the Sloan School of Management at MIT.

This paper describes a case study within a major international bank where an attempt was made to utilize *technology* to mediate the conflict of forces toward integration, demanded by the needs of new *strategic* products and services and to respond to competition, and forces toward autonomy, inherent in the corporate culture of the *organization*.

A conceptual system architecture was developed to serve as a blueprint for integrated autonomous processing. Although still evolving, the conceptual model has been shown to be very effective at guiding system development and mediating the forces of integration and autonomy. The bank's environment, the motivation and description of this conceptual model, and the experiences of the past three years are presented in this paper.

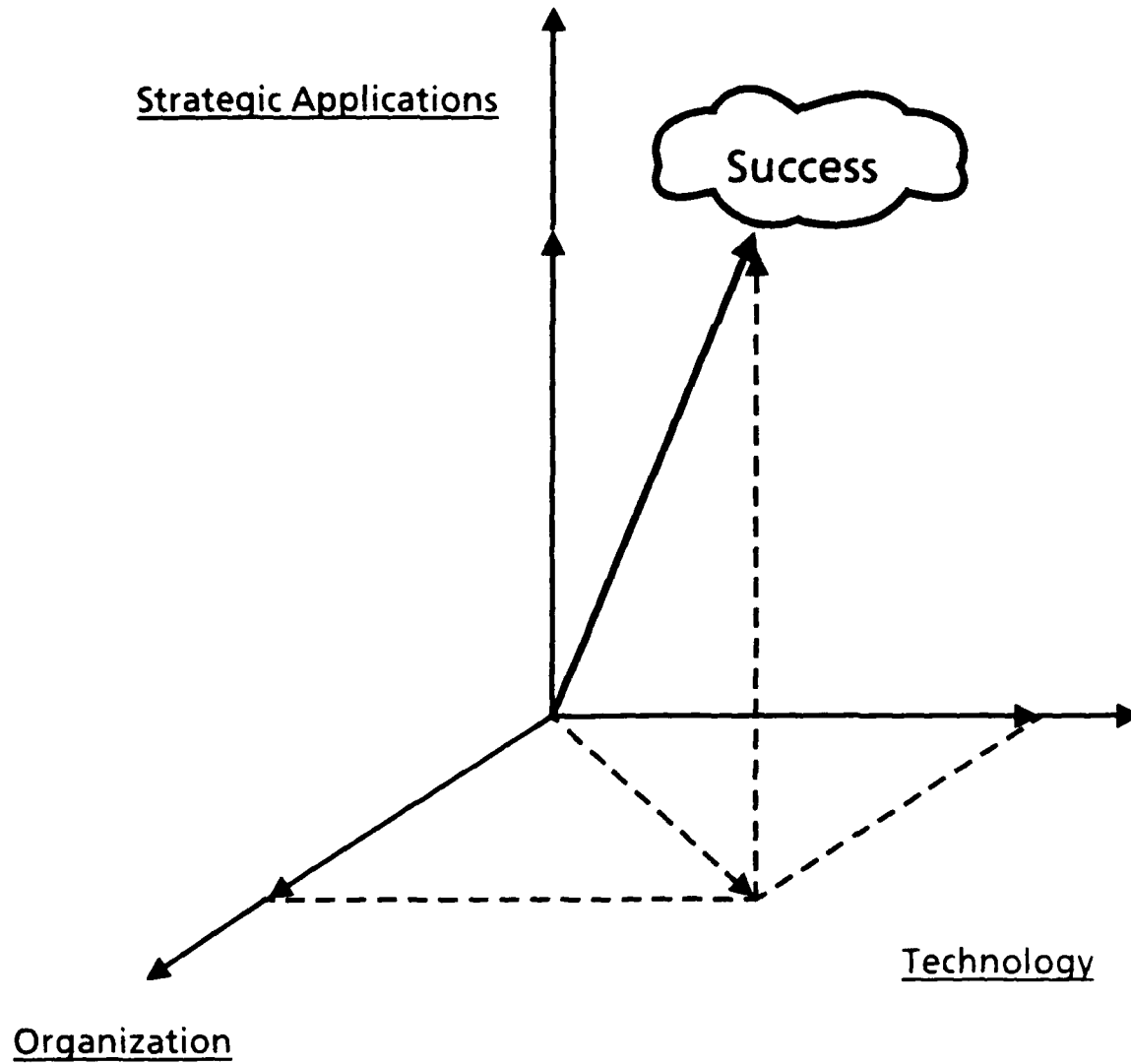


Figure 1 A Strategic Applications, Technology, and
Organizational Research Initiative (SATORI)

2. Banking Environment

The banking environment has experienced dramatic changes over the past two decades and it continues to change at an accelerated pace. Competitive pressure is increasingly being imposed on the banking industry on a multitude of fronts. There is little to constrain wholesale oriented institutions from crossing interstate barriers. Competition is also crossing international frontiers. In many countries, international wholesale banking is becoming much more aggressive as banks vie for the business of multinational corporations.

Information technology is being used extensively in banking [Lipis, 1985]. For example, in 1986 the value of computerized payments processed by the New York Federal Reserve and New York Clearing House often exceeded \$1 trillion in a single day. Technology has been critical to keep pace with the increased volume of financial activities; payments processed through New York financial institutions have increased 50-fold in the past 20 years, to the point where every four days an amount equal to the total annual US GNP is turned over.

During the 1990's, technological innovation will have an even more dominant effect on the financial services environment than interest rate volatility. The development of sophisticated information processing facilities are enabling institutions to offer a much broader range of products and services on a global basis with greater efficiency. Technologies such as distributed systems and database machines [Hsiao and Madnick, 1977; Madnick, 1977] are being employed by the industry to gain strategic advantage [Keen, 1986; McFarlan, 1984]

2.1 Autonomy, Integration, and Evolution

This paper reviews and analyzes the development and deployment of a conceptual model for integrated autonomous processing in a major international bank. The cultural forces in the bank favored an autonomous non-integrated approach: a tradition in which the responsibility for getting a job done is itself distributed to the lowest possible level; to maximize the independence of projects. To cope with the constantly changing environment and the distributed autonomous culture, it was recognized that three key goals need to be satisfied in designing information systems: autonomy, integration, and evolution.

On the one hand, each bank product (e.g., funds transfer, letter of credit, loans, cash management) is developed autonomously. Each product manager, in general, has complete freedom over his hardware acquisition and software development choices (e.g., hire his own development staff, retain outside contract programmers, or buy/modify suitable application packages). In the culture of this bank, this autonomy is critical since each manager is held solely responsible for his products. Excuses such as "the data processing people didn't do it right" are not acceptable. When information must be exchanged, it was often accomplished by "tape hands-off", usually at night, as depicted in Figure 2.

On the other hand, the needs for integration have been increasing rapidly both at the user level and database level. Since each system had its own directly connected terminals, users that required access to multiple systems had to have multiple terminals in their office, or walk to an area of the building that had a terminal tied to the system needed. The "tape hands-off" mechanism was used to create "shadow"

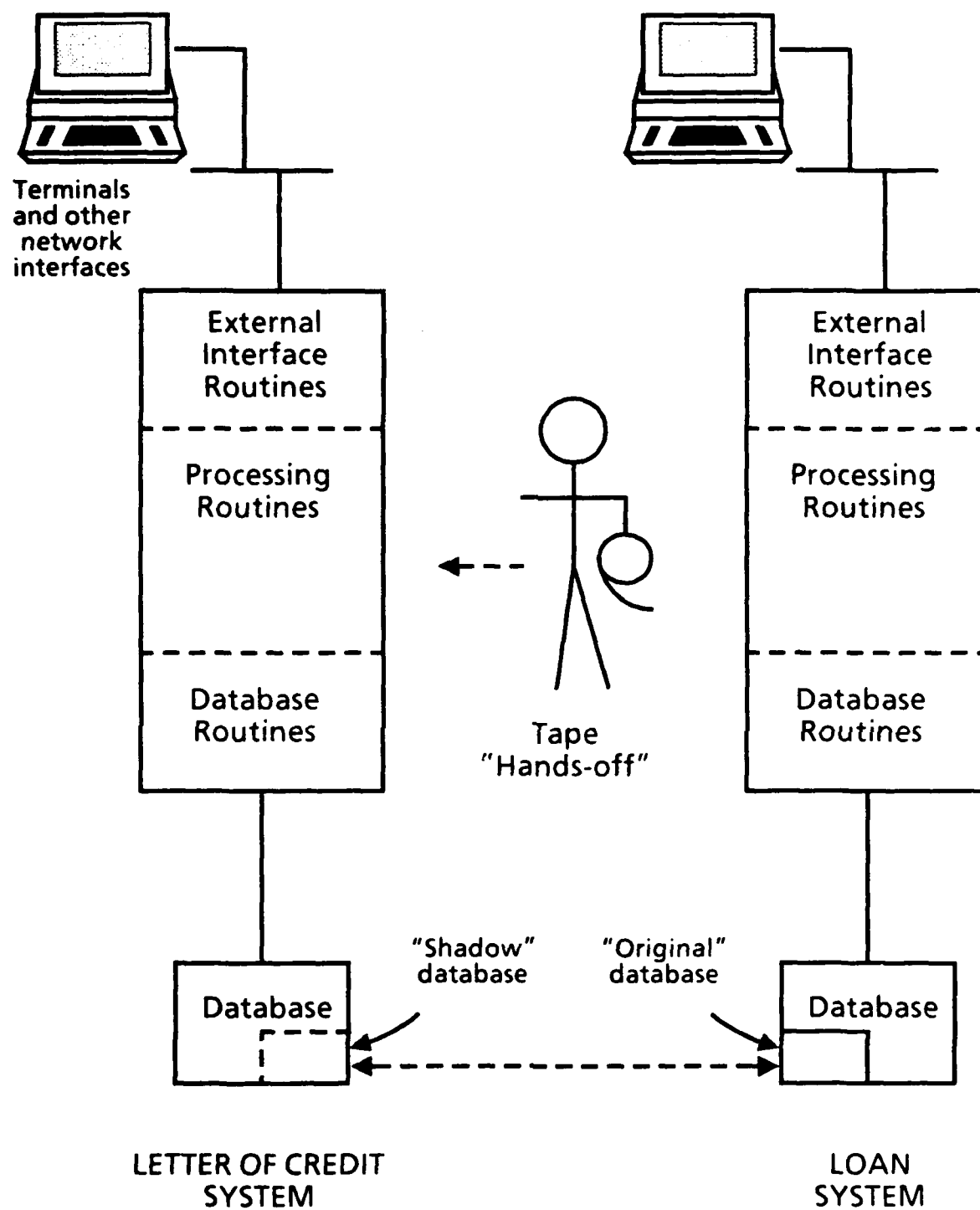


Figure 2 Independent Autonomous Systems

databases of each other's real databases. Since the shadow database diverges from the real database during the day, inconsistencies could result.

The problem of integration has been intensified by the need for evolution in at least three areas: current products, new products, and new technology. As the current products become more sophisticated, there is need to acquire more information from other systems. Increasing "tape hands-off" leads to processing complexities and does not address the need for up-to-date information. Many of the new products (e.g., cash management) are, in fact, a repackaging of existing products. To produce completely new systems would be expensive, time consuming, and impractical. Finally, to maintain an efficient and cost-effective environment, it is important to be able to take advantage of new hardware components without disrupting or discarding existing systems.

Traditional centralized database management system strategies provide integration, but have limited capabilities for evolution and autonomy. The purpose of this paper is twofold: 1) present a conceptual model, based upon [Lam and Madnick, 1981; Madnick and Wang, 1987], that describes the architecture for an evolutionary, integrated, and autonomous environment; and 2) report the experience to date in implementing this model by the institutional banking group of a major international bank.

During the past three years, five products have been implemented using this conceptual model. This paper focuses on two of these systems: transaction investigations and management information system (MIS) reporting.

2.2 Transaction Investigations

Complex international financial transactions performed in high volume can be error prone, resulting in a significant number of discrepancies between customers' records and bank records. Customers inquire about the source of these discrepancies and the bank is responsible for reviewing its records to establish what it did and why, supplying these records to the customer, and making good if the bank is in error. The bank's activity with respect to such an inquiry is called an investigation.

Investigations are the job of sizable staffs in a large bank. They are usually performed with the aid of data stored on the transaction processing systems, microfiche, and printed reports. In addition, the investigation activity itself generates a history, which should be tracked to produce indicators of productivity, customer satisfaction, and the efficacy of the transaction processing systems. The Historical Data Base (HDB), needed by the bank to support these investigations and other applications covering the previous 90 days, must be able to hold at least 40 million records.

2.3 MIS Applications

Each transaction processing system generates various summary reports. Periodically, integrated MIS reporting for upper management was accomplished by manually entering data from a variety of these reports and other sources into an Apple computer spreadsheet.

Management desired new and more comprehensive MIS systems which integrated previously independently generated and inconsistently presented data. In total, a database of at least 12 million records, with over 250,000 additions per day, is needed

to hold the relevant data. Furthermore, these new systems should facilitate the modelling of the effects of alternative decisions (i.e., what-if analyses fed by current performance data).

3. Conceptual Model

3.1 Model Architecture

A flexible system architecture was conceived to serve as a "blueprint" or "conceptual model" to guide the development of new systems. This conceptual model consists of seven major functional components as depicted in Figure 3. These components are separated into five layers, with the integrating application-independent layers (external interface, message control, data control, and shared data resource) surrounding the application processing components [Madnick and Wang, 1987]. For this bank, these application processing components are separated into three classes of applications: transaction processing, information processing, and administrative processing.

3.2 Autonomy Aspects of Model

This architecture attempts to mediate the conflicts between the goal of autonomy and the goals of integration and evolution. All of the layers, except for message control and data control, lend themselves to unlimited autonomy. Each product manager could acquire and manage his own resources including 1) terminal/network gateway hardware, 2) application processing computers and software, and 3) database computers and software.

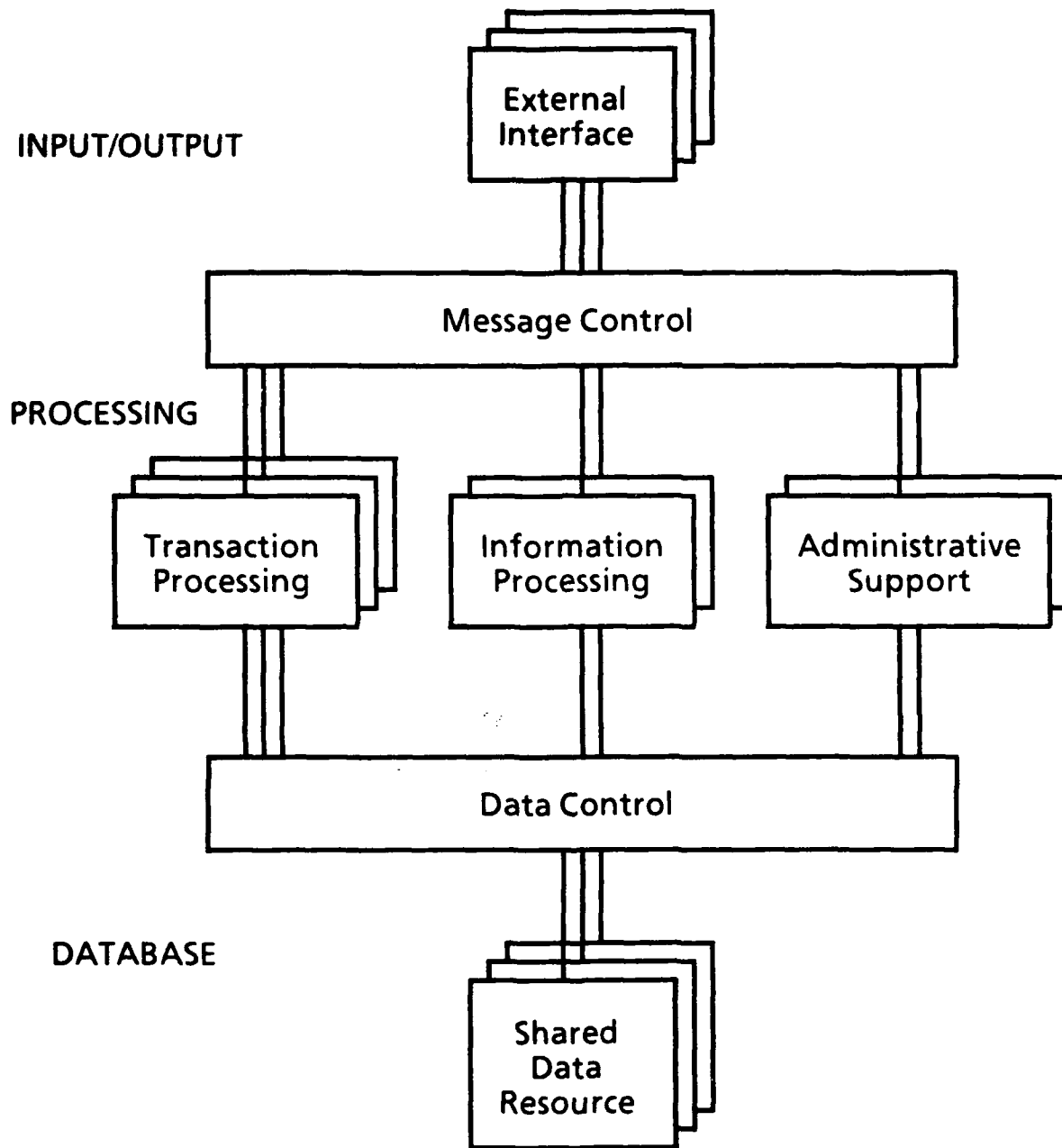


Figure 3 A Conceptual Model

In the past, as shown in Figure 2, these three decisions were bundled together. In practice, the primary concern for autonomy involved the application processing, with a lesser concern for the database, and minimal concern for the external interface. It is in the application processing that the functionality of the product is manifest. It is important that enhancements and corrections, as well as the initial development, be able to proceed with minimal needs for coordination or delays due to the managers of other areas of the bank and other computer systems.

Given the architecture in Figure 3, each manager has complete control over his application processing system. Furthermore, as many separate database systems as needed, or desired, can be created. It is expected that initially there will be more databases than theoretically needed, due to the influence of past practice. The integrated autonomous architecture provides access to these databases by other applications as well as an evolutionary path for eventually integrating these databases, as the needs for integration intensify.

3.3 Integration and Evolution Aspects of Model

There are several underlying concepts and components of the model which address the issues of integration and evolution.

Message Control and Data Control perform unifying functions for the model. They are the points at which all processing will be coordinated. For example, in principle, any terminal can access any application. Furthermore, application subsystems can utilize the Shared Data Resource to manage and maintain data which is common to more than one component.

Message Control and Data Control are both conceptually single entities. The other components are types of processing functions. There may be many instances of each

type (e.g., multiple transaction processing systems and multiple shared data resources).

3.4 Model Components

It is important to realize that the model components of Figure 3 are logically separate. They could be mapped to actual hardware in various ways. For instance, in the transaction-investigation system, each component resides on a separate processor; whereas, in the MIS system, many components reside on a single processor.

3.4.1 External Interface

The external entities interfacing with these banking systems fall into five categories: 1) payment networks; 2) communication networks; 3) customer terminals; 4) professional workstations; and 5) other intra- and/or inter-bank systems.

3.4.2 Message Control

Message Control coordinates the passage of messages between the processing components. This involves routing, translation, sequencing, and monitoring:

- o Routing accepts a request for the delivery of messages to a particular logical function and determines the appropriate physical address to which the message should be delivered. Routing can thus accommodate changes in the availability and location of functions. Routing can also help coordinate requests for services that involve several functions.
- o Translation maps a limited number of protocols from one standard to another.

- o Sequencing determines the order in which messages are to be delivered to recipients on the basis of established priorities.
- o Monitoring determines the state of messages within the system at any given time. Monitoring thus includes the responsibility for the integrity of a message from the time it is presented by one component until it is accepted by another.

3.4.3 Transaction Processing

Transaction Processing refers to the applications which execute the customer's financial instructions. These systems typically retrieve and update a significant amount of data (e.g., client balances). The sub-functions of Transaction Processing include validation, risk management, and accounting.

- o Validation functions are those which perform review of instructions to ensure that all information needed for processing is present and that the information is consistent with previously defined rules for data elements. Incomplete or invalid requests may be "repaired" by augmenting or clarifying the request information, either with information available internally or from external sources such as the requestor.
- o Risk Management functions are those which verify that the transactions being processed do not violate limits, conditions, or policies established by the customer, the bank, or the various regulatory agencies. Ideally, these functions should be synchronized with the processing of transactions. In some cases where this is not feasible, "after-the-fact" Risk Management functions can be used to initiate corrective action.
- o Accounting records the cumulative impact of the transactions completed. Accounting functions are characterized as being continuous over time, in contrast

to the discrete events which take place in most of the transaction processing functions. For example, in the banking environment, accounting takes place on two distinct levels: customer accounting and organizational accounting.

3.4.4 Information Processing

Information Processing refers to all the subsystems that perform analysis, calculations, or restructuring of the data (e.g., consolidated financial statement). The sub-functions of Information Processing include user interface, static reporting, ad hoc reporting, and access to outside data resources.

3.4.5 Administrative Support

Administrative Support provides facilities for the performance of office functions by administrative or managerial personnel. This activity is required to maintain organization, procedural or personal information. Example facilities include electronic mail, word processing, correspondence files, and inventory controls.

3.4.6 Data Control

Data Control coordinates access, presentation, and the passage of data between processing functions and the Shared Data Resources. It routes queries and updates to the appropriate component of the Shared Data Resources, performs security and priority functions, maintains concurrency control over the shared data, and returns responses to the appropriate processing function.

3.4.7 Shared Data Resources

Shared Data Resource is the component responsible for holding the information common to one or more other components of the Model. Although this activity is logically centralized in the Shared Data Resource, it may contain multiple elements

(storing different segments of the shared data, or different organizations of the shared data). The Shared Data Resource performs two functions:

- o Information Management determines what information must actually be stored and retrieved to satisfy the request, performs the transformations necessary to produce the required information, and determines how the information is to be stored or retrieved.
- o Storage Management determines physical locations of data and access storage storage devices.

4. Implementation Experience

4.1 Summary of Results

The conceptual model of Figure 3 has been the general architectural guideline for system development over the last three years. Two particular projects completed in that time, including two large databases (20 gigabytes and 1.5 gigabytes), will be described. VAXCLUSTER technology and the ORACLE relational database management system was used extensively to implement the conceptual model. A portion of the development of one of the applications used the STRATEGIM language. Most of the other applications were either programmed in the C language or used existing packages.

The goals which led to the development of these two systems were threefold: 1) to create effective applications for particular user groups; 2) to provide a central repository and further dispatch point for all banking transaction processing historical data; and 3) to provide an application systems architecture in which MIS data of various kinds was organized according to a partial order of increasing levels

of abstraction or aggregation, so that higher levels of data would be created by batched flows of data from lower levels.

The design and implementation of the systems to meet these goals were greatly affected by the cultural factors and business considerations mentioned earlier. The first and last of the goals (good applications and structured aggregation) have been largely achieved by the implementation. The second goal (shared historical data resource) was approached very cautiously. New technology and the experience gained makes full attainment of this goal more feasible today than it was in 1984.

Despite much skepticism about the performance of a relational database system on databases as large and active as these, particularly on minicomputer technology, the applications making use of these databases ultimately performed very well.

4.2 System Design

4.2.1 Application of the Conceptual Model

General plans for new systems called for the segregation of system development efforts, and of hardware and software components, into specific classes of systems which would correspond to components of the conceptual model.

It was important that this be done on an evolutionary basis since the inventory of existing systems constitutes some 20 million lines of code. There was (and is) insufficient business motivation to replace all these working systems even to achieve such general goals as data integration. Instead, as new applications are built and old ones replaced, it is intended that they be brought into greater conformity with the model. This means that only those pieces of integrative components required to support a developing application may possibly be built.

In addition, it was not practical that the integrative software and hardware required for Message Control and Data Control be custom built for the bank. Instead, it was expected that these components would be purchased commercially. At the time the model was proposed and accepted as the basis for new development in the bank, complete Message Control and Data Control software was not commercially available. Over time, it has increasingly become available. Thus, one of the major values of the model has been positioning the bank for the arrival of such new technology.

The following two sections describe the ways in which the ideas of the conceptual model were applied to the development of the historical database system for transaction investigation and the related MIS system, first from the point of view of their functional organization, and then from the point of view of implementation issues.

4.2.2 Role of a Historical Database

The historical database was envisioned as a data resource shared by multiple applications: Transaction processing systems would no longer have the responsibility of storing historical data nor need to produce a variety of different summaries of the same data for the different information processing systems. Instead, all completed transactions and proofed account balances of each type would simply be written to a common database. Information processing systems would each extract the data they needed from the historical database, and use that data independently from the uses put to it by other systems.

The goals of such a historical database was to simplify the work of transaction processing systems, and more importantly, to simplify inter-system data flows. The

current flows had reached a level of complexity which were, in total, no longer known to any one person and had no discoverable principle of organization. They required several people several months to catalogue. Of most urgency, dependencies on tape hands-off and other information flows between systems meant that each year more systems were unable to complete their off-line work in the daily time periods available, especially because customers all over the world demanded that some systems be available on-line virtually all the time. The data-flows to be provided by this new scheme are shown in Figure 4.

4.2.3 Structured Aggregation in MIS Systems

In addition to the benefits mentioned above, the integrated MIS system was intended to improve the overall organization of inter-system communications through a systematic way which MIS applications should communicate with which other applications.

This was to be accomplished by describing the input requirements and the output of various desired MIS applications in a uniform way, and then identifying the lowest cost (least transformation required) connections between such applications, so that the output of some applications becomes the input to others.

The output of each MIS application is treated as a database in its own right, and the data-aggregation function is regarded as the primary role of this sort of MIS application (e.g., standard cost accounting). The result is a partial order of aggregated databases in which all databases depend ultimately on the raw historical data, as depicted in Figure 5. The current collection of MIS databases consists of seven levels, and the dependency diagram barely fits on a large wall.

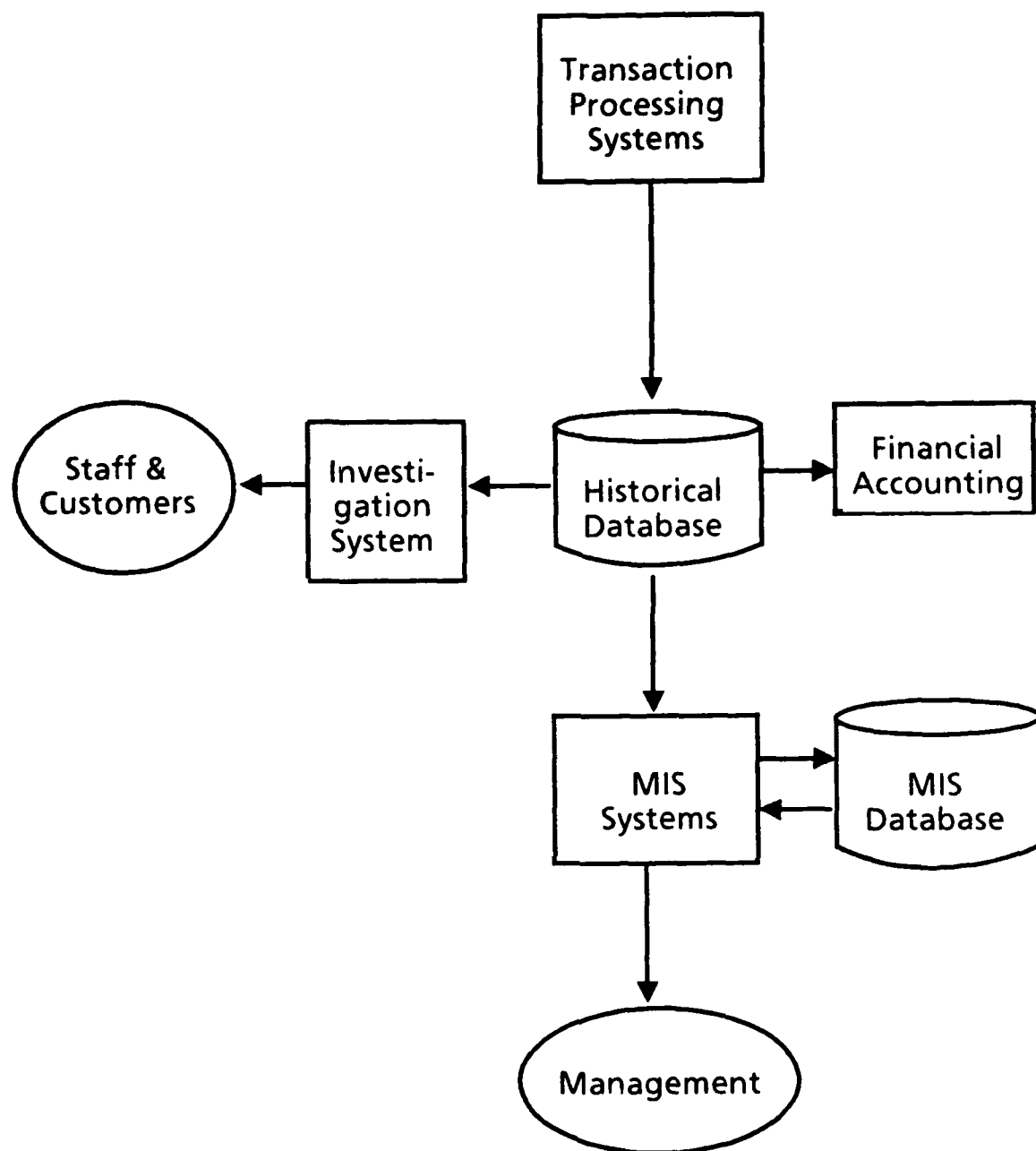


Figure 4 Role of Historical Database

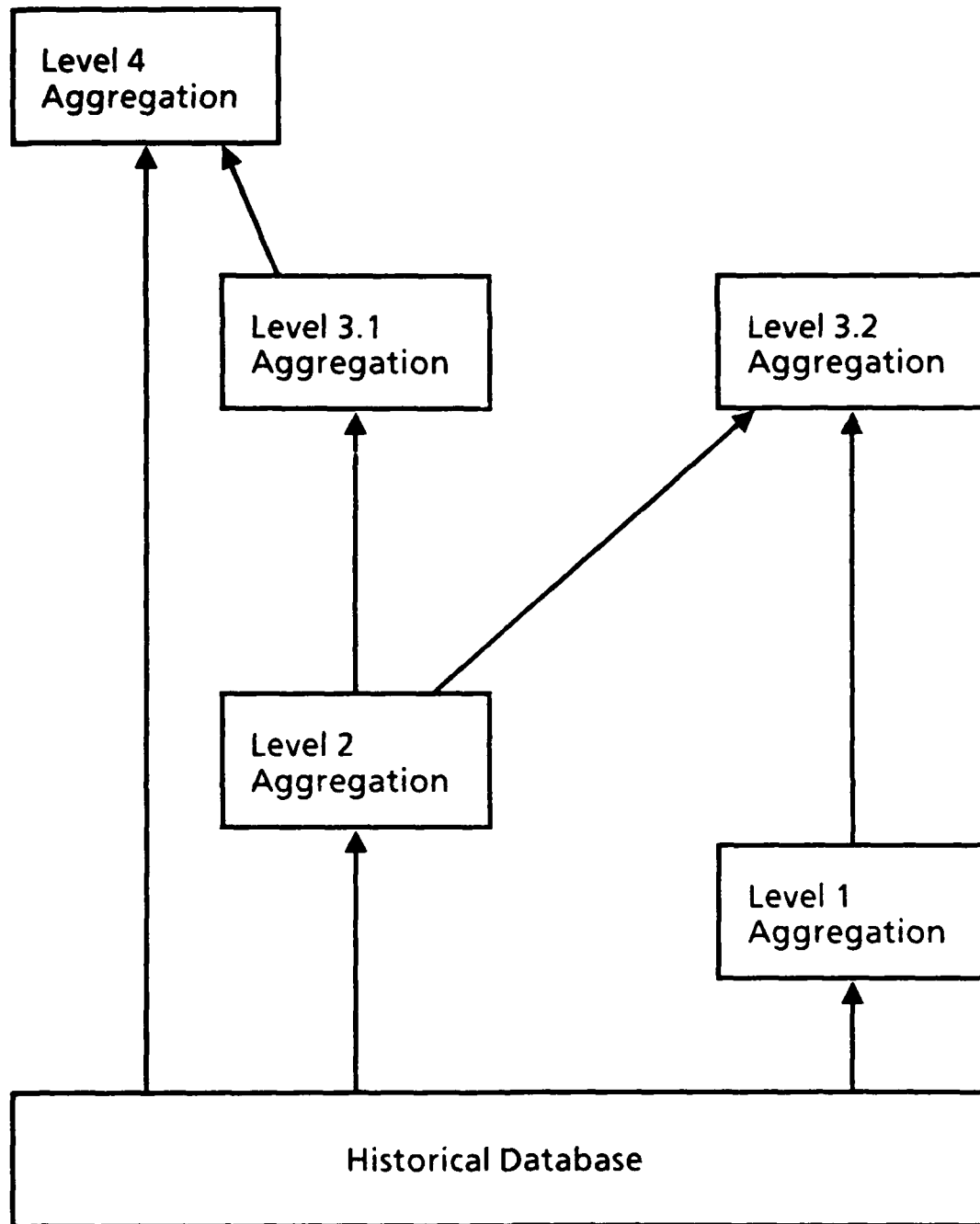


Figure 5 MIS Data Partial Order of Aggregation

The general design of the MIS system is to regard various sets of tables as levels of aggregation. Although the processing is quite complex, the fact that all of the data is maintained as a single shared data resource has dramatically simplified the operations of the MIS system components.

4.3 Model Interpretation

By "interpretation" we mean a functional mapping of model components to hardware and software components. The interpretation of the model has varied considerably between the transaction processing and information processing environments, and has varied over time. For instance, front-end processors are now being distributed geographically, which has caused some changes and extensions.

The initial interpretation involved a "null" message control layer, since the early information processing applications did not communicate with each other, and all screen management was done in the same processor as the applications, although with separate software. Only later, when screen management software capable of communicating via datagrams with the applications processors became commercially available on personal computers, did a message control layer become significant in information processing.

The five layers of the conceptual model are interpreted as shown in Figure 6: (1) The *external interface* system consists of terminal controllers, wide-area gateway boxes, a terminal to host ethernet, network interfaces for the application host machines, and the screen management software on these hosts. (2) The *application layer* consists of the application software. (3) The *message-control* layer shares a host-to-host Ethernet with the data-control layer. (4) The *data-control* layer consists of software

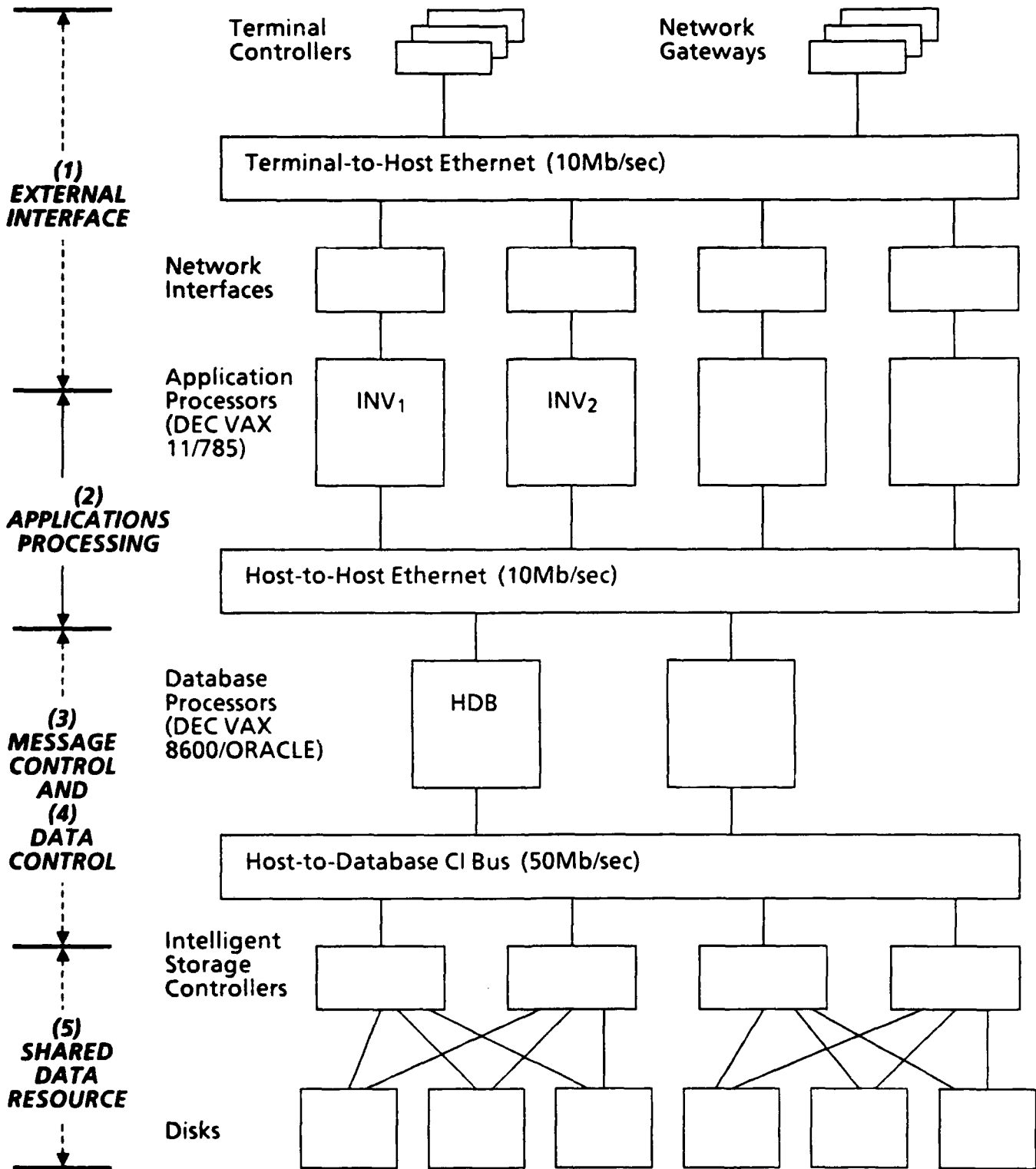


Figure 6 Model Interpretation

on the application hosts enabling communication with the database processors over the host-to-host Ethernet, the communications software on the front-ends of the database processors, and a portion of the database management system which includes query analysis and concurrency control. (5) Finally, the *shared data resources* layer consists of the back-ends of the database management systems running on the database processors, and the storage systems to which they are connected. These components are described in more detail below.

4.4 Systems Software Requirements

The bank's development policies emphasize individual projects be as small as possible, both in number of personnel and time frame. Ideally, a project should take less than a year and require no more than five developers [Appleton, 1986]. The systems software chosen to support these information processing applications differed from that chosen to support transaction processing applications (such as the funds transfer system). The major reasons for this difference are the technical and performance differences between the transaction processing and information processing systems.

The information processing systems communicate with the outside world largely interactively. Their databases are the only components which must be fully recoverable. The tolerable time to recovery may be as long as several hours, whereas, for transaction processing systems this is typically a matter of minutes or even seconds. In addition, information processing systems are highly fluid: requirements change from month to month.

The primary system software required on the information processing side were flexible screen managers, database management systems, and host-to-host communications software. These systems lend themselves to prototyping and non-procedural languages.

4.4.1 Operating Environment

The hardware and software chosen to realize these components was largely based on the DEC VAX. The group's processors currently include about thirty VAXes and two IBM mainframes. While the IBMs exchange messages with the VAX systems, the two environments are not currently integrated into the same conceptual model: the IBM systems, developed earlier, perform high volume batch work running purchased packages which provide end-to-end support for these applications.

The reasons for this preponderance of VAXes were the experience base of available developers and support personnel as well as the cultural forces, noted earlier, favoring separate computers for separate jobs. Thus, VAXes have invariably been chosen in the cases where preliminary analysis indicated that VAX systems were capable of handling projected workloads, and where specialized software was not immediately available off the shelf for the IBM MVS systems.

In the case of the transaction investigation system, the availability on the VAX of a software package to support investigation tracking was an important factor in the selection of the VAX, since here the size of the database might have indicated that a mainframe would be a better prima facie choice.

As a result, VAXCLUSTER technology has been widely used in a variety of ways. To keep Figure 6 simple, certain complexities have been omitted. For instance, some of the applications processors are also connected to the Computer Interconnect (CI) Bus

to allow them to take the place of a defective database processor. Furthermore, for increased reliability, each Ethernet shown has an additional backup Ethernet.

4.4.2 External Interface

All communications with the ultimate system users is terminal-based. Certain data are downloaded to PCs for spreadsheet manipulation, graphics, and report printing. Terminal-to-host and PC-to-host communications (part of the External Interface system) is accomplished physically by an Ethernet. In fact, two Ethernets are used: one for production systems and another for development systems which serves as backup for the production network.

Bridge controllers (local terminal controllers, dial up-systems, and X.25 gateway controllers, which provides a connection to remote terminals via an in-house worldwide network) are used to serve both the terminals and the computers. They communicate with each other via XNS. Sun Workstations are used on this network as network control and configuration management devices.

Screens are managed from the VAX application machines, using Viking Forms Management software, which in turn communicates with the application software.

4.4.3 Applications

The transaction investigation application software largely consists of calls to the screen management and the database management systems. In addition, it performs any processing required to make the necessary transformations between these two systems. The package purchased to support these investigations, in actuality, does considerably more than this.

4.4.4 Data Control

The Data Control level consists largely of database management software and general purpose hardware. Some of the features of Data Control could not be implemented, owing to the lack of commercial distributed database management systems at the time of implementation.

In addition to development productivity, the major issues concerned performance, because of the large size of the databases. Other important criteria were: 1) preferences for relational systems as having the longest future and offering the greatest productivity gains, 2) relatively widely used systems, and 3) adherence to standards and ad-hoc standards when possible, such as CODASYL or SQL.

At the start of the project, there was considerable, mostly unsolicited, expression of opinion and concern that relational systems "performed poorly" and were therefore unsuitable for large databases. Of course it is meaningless to talk about good or bad performance except with respect to a specific use and to specific parameters of performance. In this case, both the historical databases and the MIS database would be written almost solely in batch mode. Although the volume of such writes was quite large (300,000 512-byte rows per night), the writes were all appends, no update; reads and writes were never expected simultaneously against the same tables; all reads were predictable since very little access to the databases would be ad-hoc; and the database could be designed in such a way that most read requests would require no joins between tables.

Performance projections were established by study of the results of benchmarks performed by other institutions, by review of published performance analyses,

including a study by the National Bureau of Standards, and by analysis of the likely effects of design features of the systems.

It was quickly established that the commercially available systems based on pointer chains (which happened to coincide with the available CODASYL systems under consideration) were incapable of performing the nightly number of batch appends even within a period of 24 hours, while the systems based on indexes (the relational systems under consideration) would permit these batch appends within a number of hours.

The two systems considered most seriously were INGRES and ORACLE. Benchmarks as well as general opinion suggested that ORACLE's performance in performing unlocked reads against single tables in large databases was superior to that of INGRES. ORACLE exhibited linear (and almost flat) increase in response times to such simple queries as the size of the database increased. INGRES was superior to ORACLE in the performance of complex joins, and offered superior fourth generation development tools.

For the type of production applications envisioned, ORACLE appeared to be a better choice, although neither system would be, in its then current version for VAX hardware, capable of handling the number of simultaneous users ultimately projected for the investigation system. Special design approaches were used to overcome these problems. Furthermore, one could be confident that both hardware and software performance would be continuously improving.

4.4.5 Shared Data Resource

The data resource used in all the VAXCLUSTER systems is supported by DEC cluster storage, which consists of the very high speed CI Bus (70 megabits per

second), connecting a number of processors and a number of intelligent controllers (currently a total of 16 such devices) in such a way that any processor can communicate with any controller. Controllers are in turn connected in pairs to dual ported disks, allowing full redundancy of hardware components. For instance, within the cluster, when a processor fails, its work can be taken over by another processor which has immediate access to all of the same disks.

4.5 Development Histories

4.5.1 Changes in the Plans

In developing these systems, it became clear that the strength of certain features of the bank culture was even greater than anticipated: independence and competitiveness between managers, and a predilection for achieving fast tangible results. Steps to integration actually needed to be smaller than the original plans called for. In building and supporting stand-alone applications, each development group has the total responsibility for delivering an application to users, and each group has a strong aversion to relying on cooperation from other groups, not equally responsible for some deliverable.

Only a portion of the transaction processing data (that involved with funds transfers) was required to support the needs of the investigation application. The MIS system, however, required summary data from all the transaction processing systems. These systems already had the capability of providing the lowest level of aggregation required by the MIS applications, since they were supplying this data to current applications. A coordinated effort would therefore provide theoretical future benefits, while increasing current development time, cost, and risk.

As a result, the transaction investigation and MIS projects, which were started at approximately the same time, wound up with each largely going its separate way. That is, instead of extracting and aggregating data from the historical database produced by the transaction investigation system, the MIS system receives its own partially aggregated data feeds from transaction processing and financial accounting systems, as shown in Figure 7.

4.5.2 Transaction Investigation System (HDB)

Projections called for at least 100 simultaneous users using the investigation system and having read-only access to its historical database (HDB), with maximum rates of 1 database request each five seconds.

The applications and the database are supported on a VAXCLUSTER, as shown earlier in Figure 6. A VAX 8600 is used to support the database, while the investigation application runs on multiple VAX 11/785 "front end" machines, which communicate with the database machine via DECNET and the VMS mailbox facility. The investigation application constructs a query in the form of a "query type" and a set of parameters. The historical database system translates this into an SQL query, and returns the response table to the application.

The configuration has grown to two front-end 11/785s and an 8600 "data base processor", while ORACLE has gone through two major new releases. Several changes in design to improve performance have also been made over the years. The modular evolutionary capability of the conceptual model architecture greatly facilitated all of these changes.

The initially installed production system (on a VAX 11/780 front-end with a VAX 11/785 back-end) supported 30 users with database query response times of 90 to 120

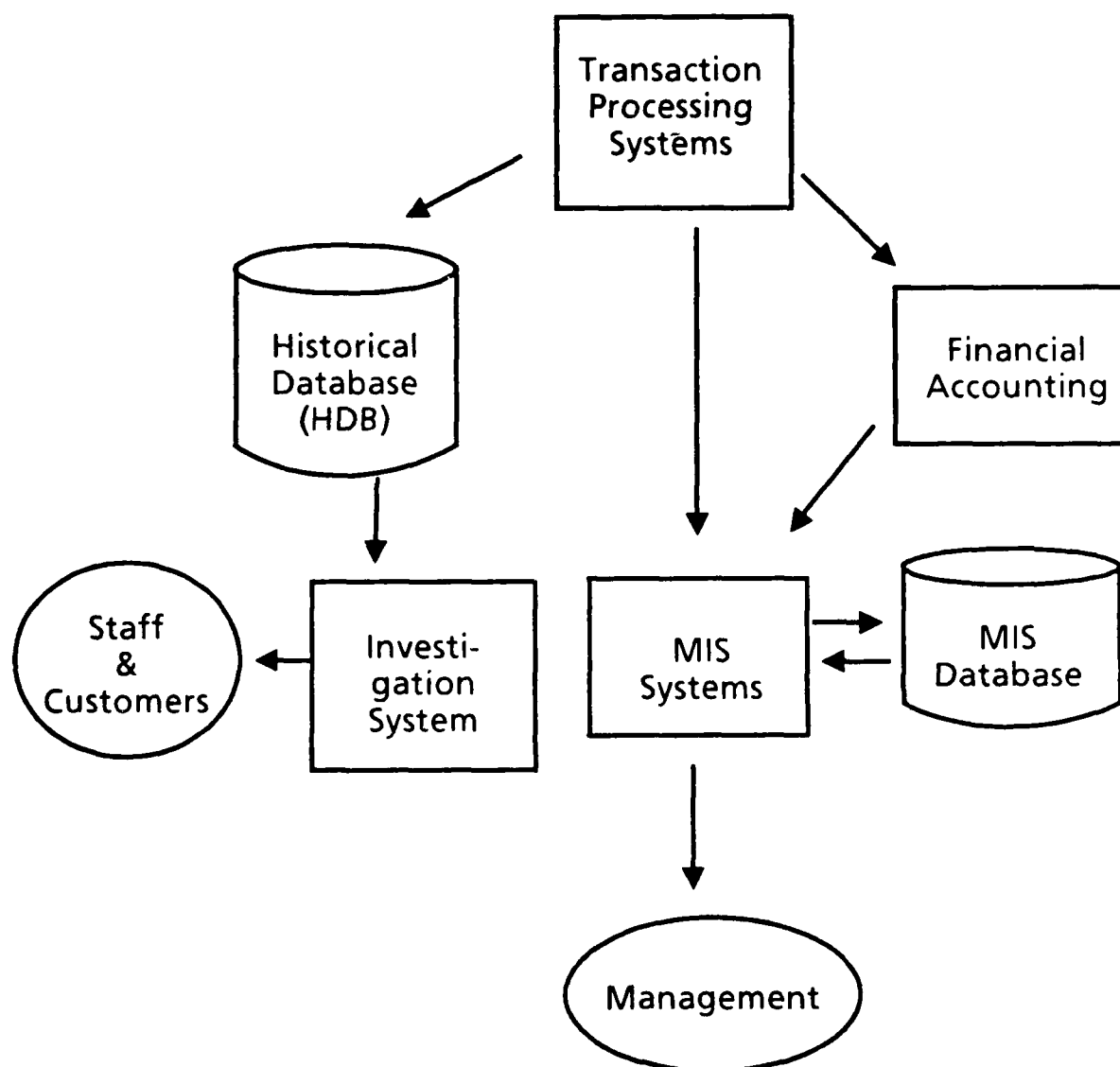


Figure 7 Current Data Flows

seconds. The current software and hardware supports 160 simultaneous users with database response times of 5 to 7 seconds. Before automation, each search for historical data required at least 15 minutes of an investigator's time.

The full cycle of design, development, testing, installation, training, and live operation of the transaction investigation system required about six months, with one programmer responsible for the software development.

A significant number of small problems, and a few large ones, were discovered in the course of development. For example, it was discovered during the course of development that the ORACLE "money" data type would not hold large enough amounts for the needs of such a very large bank. It was also discovered that the tape recovery of very large tables would take much too long, and that mirroring was not economically justifiable. It was also found that the re-indexing of very large tables was taking too long (the original analysis counted only the time required to index the newly appended data each day, not the entire 20 gigabyte database.) To solve this problem, tables were partitioned by dates, and application code was written to permit searches across ranges of dates.

Many of these problems could have been anticipated by careful up-front studies. However, such studies would have required as much time as building the system did. Without the flexible and powerful environment used, the same sorts of unanticipated problems would have been likely to cause development disasters, as they often had done in the past in the experimental culture of the bank.

4.5.3 MIS Systems

The initial phase of MIS system development and implementation also required about six months, with three contract developers and one manager. About one third

of this time was devoted to learning to read correctly the tapes from the various non-integrated sources of MIS data, and developing programs to map these tapes to relational tables without repeating groups, multiple record types, variable length records, etc.

A major innovation of the system was the virtual elimination of the printing of MIS reports: particular pages of reports of interest to individual managers are viewed on terminals, and printed locally if so desired.

The MIS system had intense advocates in the MIS department, who specified exactly what they wanted to see in the system. It is more common in the bank for users to ignore new systems until they are delivered to them. The system has been constantly enhanced and enriched, with the MIS group developing a great deal of skill at the use of SQL and ORACLE development tools for small applications. In fact, this system is not called the "MIS" system by most of the users, it is called the "ORACLE" system.

The summary of data from this system began to be used in business planning sessions, which lead to the desire to use the data for forecasting and analytic modelling purposes. The result was the creation of a strategic MIS system, using the STRATAGEM modelling language, which receives data from some of the highest level applications of the ORACLE MIS system. The use of this system for analytic and strategic purposes, rather than purely reporting purposes, has grown slowly but steadily.

5. Concluding Remarks

The conceptual model described in this paper has served as an evolutionary blueprint. The technology of separating the external interfaces, message control, data control, and the database components from the application processing components, provided the capability for high degrees of integration needed for new strategic applications while preserving significant autonomy inherent in the bank's culture. The lessons learned from these experiences may be valuable for the many other organizations now facing similar challenges.

References

1. Appleton, D.S., "Very Large Projects," Datamation, January 1986.
2. Hsiao, D. K. and Madnick, S. E., "Database Machine Architecture in the Context of Information Technology Evolution," Proceedings of the Third International Conference on VLDB, pp. 63-84, October 6-8, 1977.
3. Keen, P.W. Competing In Time: Using Telecommunications for Competitive Advantage, Ballinger, 1986.
4. Lam, C.Y. and Madnick, S.E., "Composite Information Systems - A New Concept in Information Systems," CISR Working Paper # 35, Sloan School of Management, MIT, 1981.
5. Lipis, A., Electronic Banking, John Wiley and Sons, 1985.
6. McFarlan, F.W., "Information Technology Changes the Way You Compete," Harvard Business Review, May-June 1984, pp. 98-103.
7. Madnick, S. E., "Trends in Computers and Computing: The Information Utility," Science, Vol. 185, March 1977, pp. 1191-1199.
8. Madnick, S.E. and Wang, Y.R. "Evolution Towards Strategic Applications of Databases Through Composite Information Systems," Working Paper #1862-87, Sloan School of Management, MIT, February 1987.

GAINING STRATEGIC ADVANTAGE THROUGH COMPOSITE INFORMATION SYSTEMS

JOSEPH L. MASSIMO, III

This research effort is the second in a series that explores the development of Composite Information Systems (CIS) in a technologically and organizationally complex environment. Since access to the details of military applications was not possible due to time and security limitations, the particular situation studied is that of a major international bank that attempted to develop a standard set of core application software to be deployed around the world.

As noted in the first report (Frank, Madnick, and Wang, 1987), the bank has a highly autonomous decentralized organizational structure. In this environment the competing forces of autonomy, integration, and evolution represent major challenges.

The primary Critical Success Factors (CSFs) were motivated by the desire for excellence in cost (i.e. be the low cost provider of information) and content (i.e., provide more timely and more accurate information) through standardized system development and use of a common core set of application software. In particular, the three goals established were to provide: (1) real-time transaction processing with functional and data commonality across countries thereby reducing development costs and increasing integrations of the bank's global operations, (2) a range of services, and (3) ability to respond rapidly to local conditions.

The forces for autonomy were particular intense since: (1) the bank did not have a tradition of coordinated planning, (2) entrepreneurial actions are key to promotion, and (3) this project was intended to encompass branches around the world in the face of local customs, differing government regulations, and poor communications.

The major technical problems encountered were:

1. Poor documentation made local modifications and maintenance difficult. In fact, in some branches software was rewritten or revised primarily a vehicle for training their staff.
2. Human communication was difficult due to the geographic range, as well as language and custom difference.
3. Range of local talent made technical approaches that were easily understood in one location very challenging in another location with less talented individuals available.
4. Poor data validation made it hard to compensate for human errors, thereby decreasing data quality.
4. Lack of a truly modular and flexible design which made it hard to respond to local requirements and maintain data quality.

These situations, in varying degrees, are likely to be encountered in any attempt at such a project. Thus, careful attention to these problems are imperative. A data-driven strategy, as suggested in (Frank, Madnick, and Wang, 1987), would significantly reduce many of these problems by simplifying the documentation and providing a more modern, easier-to-understand, modular, and flexible infrastructure with opportunities for general-purpose approaches to data validation.

TECHNICAL REPORT #12

CHAPTER I

What is CIS?

In most large, complex business organizations today, there are a number of dissimilar or otherwise incompatible hardware and software systems in place. If such a large organization is to effectively manage its business resources and continue to improve its use of information technology over a period of years, there must exist some means of getting these disparate groups of computing resources to work together. Composite Information Systems, or CIS as they will be referred to here, describe large-scale computing environments operating under a certain methodology that enable them to share and/or combine resources to provide a more integrated, efficient whole. As a result of the application of various key ideas fundamental to a CIS methodology, an organization with a great deal of time and money invested in large-scale, heterogeneous computing resources will be better able to position itself ahead of its competition, i.e. to gain strategic advantage.

A "Typical" CIS Methodology

The basic aspects of a "typical" CIS methodology as proposed by Madnick et al. [1] relate a group of technical and organizational needs to the techniques for realizing these needs. Figure 1 provides a simple example of how these relationships function.

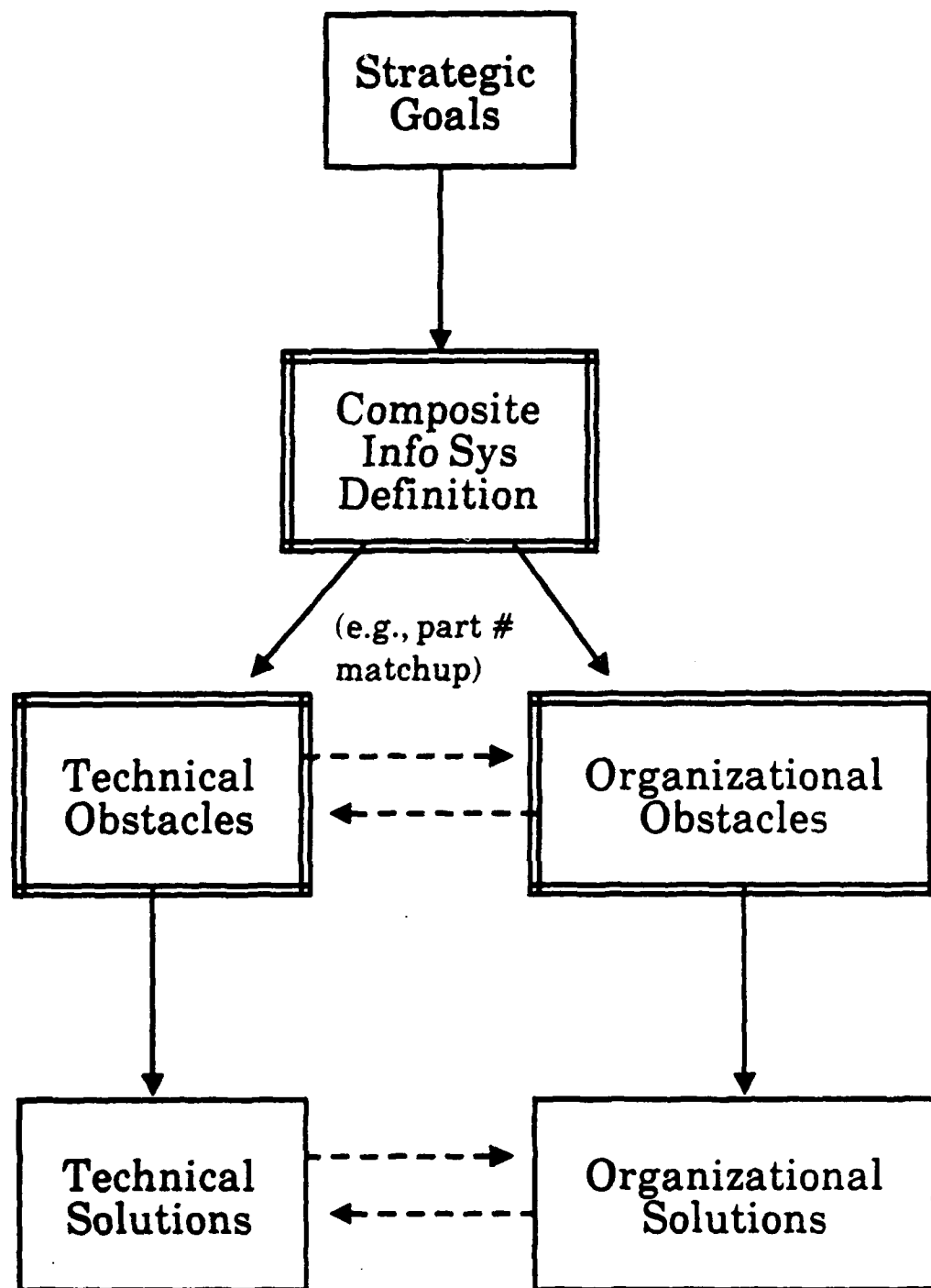


Figure 1. Composite Information Systems Methodology

The basic premises of the above methodology are: (1) First an organization must set forth and define its "top-level" strategic aims and goals; (2) The next step is to define key technical and organizational issues to address; (3) Finally, specific problems can be found that "map" onto the solutions to the key issues. In all cases, clear linkages can and must exist between what is technically feasible and what is organizationally desirable. Often, specific elements of an organization's structure and philosophy will drive a technical solution, and vice-versa. For example, a highly decentralized organization will certainly have a different perspective on the "need" for a homogeneous computing environment than one which by its very nature is more centrally operated and managed.

The Importance of CIS

Composite Information Systems, then, provide a way of looking at and defining the problem of heterogeneous computing resources within a larger context. They also pave the way to define certain conceptual models that provide a basis for "linking" together the various pieces of a large-scale computing environment. Such conceptual models must recognize the existence of different pieces of hardware, software, etc. while at the same time providing a framework to bring these elements together with some semblance of control. Although these models by definition are technical in nature, they must of necessity be implemented in practice

with regard to the structure, philosophy, and culture of the organization under scrutiny. As a result, defining a conceptual model by itself is not enough. One must also determine how the conceptual model, as applied to new or existing heterogeneous computing systems within the organization, will fit or clash with the organization itself.

Our Specific "Case Study"

In approaching the definition of a CIS conceptual model in a "live" organization, it is important to understand that the particular "fit" and/or suitability of such a model to a specific situation depends greatly on the nature of the industry and organization under consideration. The research problems and solutions presented herein pertain specifically to studies conducted at a major multinational bank. Though this bank is based in New York City, and can rightfully be considered a "domestic" bank, it also has extensive operations in a variety of countries around the world. Though the methodology of Composite Information Systems could conceivably be applied across many different areas throughout the bank, for this thesis, we have chosen to focus primarily on achieving integration across borders, i.e. applying the CIS conceptual methodology across international boundaries.

Several predominant organizational and cultural features of this bank offer potentially useful strategic applications of a CIS methodology. The most important feature of the

culture at the bank is its highly autonomous nature, and resulting decentralized organizational structure. Most business units of the bank operate independently, and managers at all levels are encouraged and expected to run their individual business units as they see fit. This is especially true at the international, or "country" level. Each country in which the bank has operations is essentially a separate "mini-bank." Top management in each country is used to a great deal of autonomy and independence from the U.S. headquarters, and in practice most units of the bank run themselves, often with very "loose" ties to and/or under little control of corporate management.

The high autonomy, need for independence, and decentralized culture of the bank have caused the proliferation of many different heterogeneous computing environments within the bank's overall information systems structure. In particular, the environments that we will concern ourselves with here are "global processing" systems, which are computing resources operating primarily across country boundaries. Global processing systems of this type afford especially interesting opportunities to study the usefulness of a CIS methodology at the bank, both because they are of necessity large and complex, and because their effectiveness within the bank directly impacts the ability of the bank to maintain and increase its overall global strategic advantage.

Nature and Scope of the Thesis

This thesis will attempt to "walk through" the process of applying a conceptual CIS model to the bank's information systems environment. We will begin by defining a particular conceptual model of CIS and its various components. We will then examine a specific existing heterogeneous global processing system within the bank, determining its strengths and weaknesses as it exists today, i.e. before application of the CIS conceptual model. The next step will be to "hypothetically" impose this conceptual model onto the global processing system as it currently exists. In other words, we want to "redesign" the existing heterogeneous computing environment so that it conforms to the dictates of our CIS conceptual model. As a result of this application of the model, we wish to determine: (1) what improvements would be brought about within the bank as a result of the model, from both a technical and an organizational perspective; (2) what detriments will arise out of the use of such a model; (3) what new problems will the application of the model be likely to cause; and most importantly; (4) in what ways would this "new" system increase the bank's resultant global opportunities and overall strategic advantage?

CHAPTER II

The CIS Conceptual Model

In determining an appropriate CIS conceptual model for our specific case study, there are a number of approaches and criteria that one may use. The primary function of such a model should be to integrate the normally separate application components into one coherently functioning whole. However, it should also take into account the need for separate subsystems to be developed autonomously. This is particularly important for the following reason: In this particular bank, as we have already seen, there is a fairly high degree of decentralization and autonomy. As a result, middle managers are encouraged to have more or less a free hand in running their separate "piece" of the business, with some degree of independence from the corporate boardroom in making day-to-day operating decisions. Therefore, when each individual operating entity must make hardware or software purchasing decisions, this is usually dictated by the local management, according to local business requirements.

Though the end result in our specific case is a large multinational organization with many different systems in place, integration can clearly not be achieved by simply requiring all local managers to "sign off" on exactly the same hardware and software systems. To do so would be in direct conflict with the culturally established tenet of

autonomy. What we see, then, is that any conceptual model designed for the purpose of integrating disparate computer systems in our particular organization must ideally be able to achieve this integration without disrupting the individualized nature of the underlying subsystems.

Conceptual Model Characteristics

What other characteristics must a conceptual model have in order to be applicable in a CIS environment? Frank, Madnick and Wang [2], in addition to the above criterion, cite several other major points that such a model should have. These points are: (1) The fact that data in a CIS environment actually may reside on many different pieces of hardware should not be noticeable to the user, i.e. data access must be transparent; (2) All users of a CIS system should be able to perform desired functions from any terminal, i.e. ability to access data and run applications should not depend on what particular piece of the system you are using. This point establishes the need for a "closed loop" in interfaces with users. (3) Existing systems not currently part of the CIS methodology should be able to be integrated without rewriting the software; and (4) The model should allow for efficiency gains between the separate systems components as a result of integration, i.e. the model should make the systems work better together than separately (this is really a fancy way of saying that the model should work!)

So given the above "rules" and goals that one should be

aware of when deciding on a specific CIS conceptual model, which one should we use? This thesis proposes to use the model established by Madnick et al. [3]. This seven part conceptual model provides for three main application components integrated via a "framework" of application-independent components. The overall structure divides these seven components into five distinct layers, as depicted in Figure 2.

The Seven-Part Conceptual Model

The seven parts of our CIS conceptual model are divided into application independent (i.e. integrative) components and application processing components. Although in a strictly "generalized" case of a CIS conceptual model, the application processing components could be defined any way we so choose, in this case we propose to use applications likely to be critical in a banking environment. Thus, the three application processing segments are: (1) transaction processing; (2) information processing; and (3) administrative processing. These are in turn surrounded by (4) external interfaces; (5) message control; (6) data control; and (7) shared data resources. The interface, control, and data resource components of the model serve to integrate and coordinate the various transaction-based, information-based, and administrative-oriented applications to enable them to function together. At this point, it would be best to further clarify the nature of each of these seven

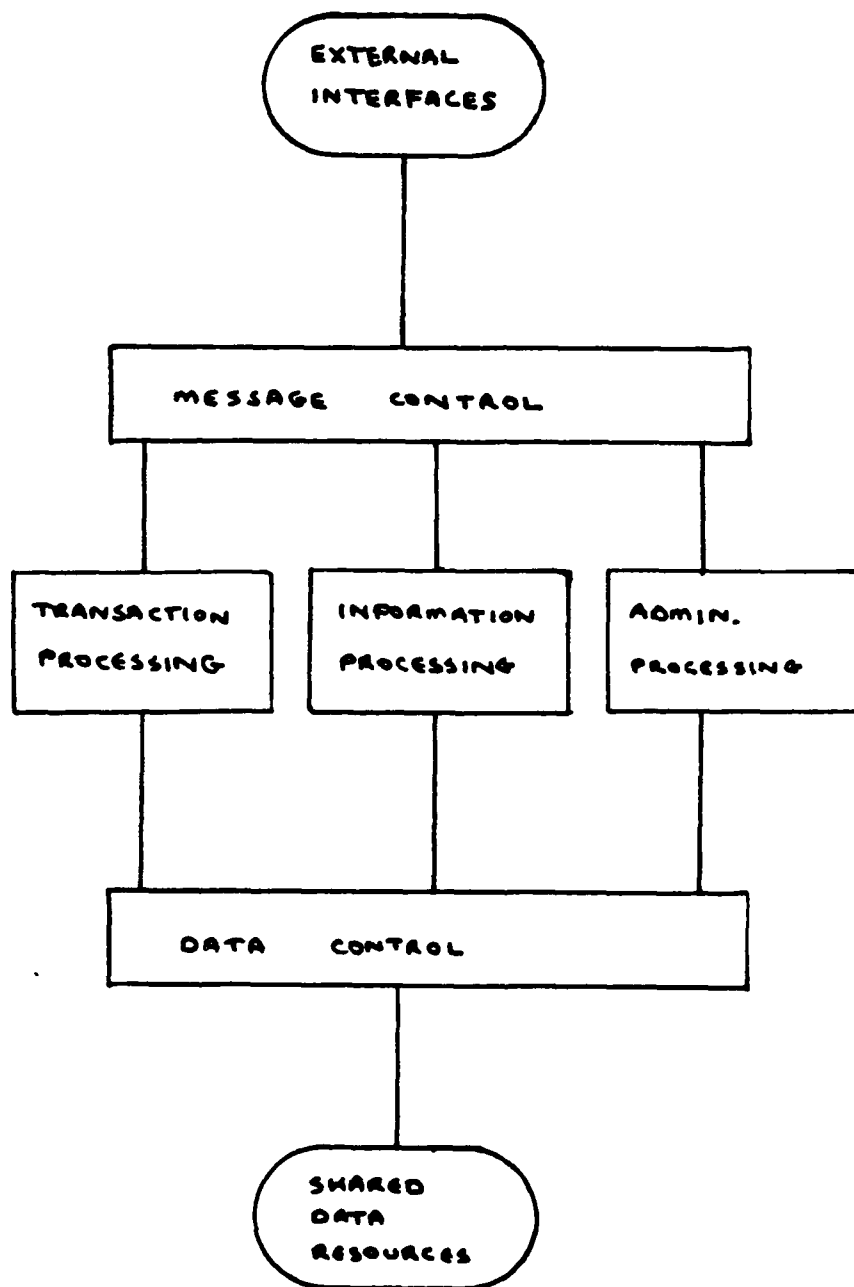


Figure 2. The Seven-Part CIS Conceptual Model.

components, in order to better understand how to apply this model in our specific case.

External Interfaces define structure and methods of communications between the CIS environment and the outside world (example: on-site customer terminals that remotely communicate with the central system). Essentially, there must exist a common means for external entities, such as users, to communicate with the CIS environment.

Message Control handles the routing, sequencing, translation, and monitoring of messages between the processing components of the model. Message control is involved actively with all three processing segments outlined above.

Transaction Processing is the first of three application processing components of the model. These applications are designed mainly to retrieve, add, delete, and update data and do very little "processing" of data (such as calculations, analysis, etc.) Some functions of transaction processing subsystems as they would apply in this particular bank include: data validation, accounting of data (e.g. audit trails), risk management (e.g. verifying that data being processed is "feasible"), and routine recording or "capture" of data as it passes through the system. Some typical examples of a transaction processing application would include updating a customer's account balance, verifying a credit limit, or providing an audit trail at fiscal year's end.

Information Processing applications are designed

primarily for performing analysis, calculations, or data restructuring. Some typical applications of this nature would be a spreadsheet, a sales projection, or a consolidated financial statement.

Administrative Processing functions provide general office automation functions for the support of managerial and administrative personnel. Such functions typically include: word processing, electronic mail, document storage and retrieval, control functions such as activity reporting, and graphics capabilities.

Data Control coordinates and controls the communication and passage of data between the above three application processing functions and the shared data resources. Five primary characteristics are essential to the implementation of Data Control: (1) Security, which ensures that individual user access to the shared data resources can be monitored and controlled; (2) Data Presentation Methods must exist to provide for query capability as well as a common Data Definition Language (DDL); (3) Routing capability must be provided so that requests for data can be sent to the appropriate database or databases, and the results returned to the proper application; (4) Sequencing permits multiple requests for data to be prioritized in such a way as to minimize overall response time; and (5) Concurrency Control makes certain that multiple applications and/or users do not alter the same data elements, thereby maintaining the integrity of the shared data resources.

Shared Data Resources hold the information required by one or more other components of the CIS conceptual model. Although the Shared Data Resource (s) can theoretically be a single database on a single piece of hardware, in practice there are usually several discrete databases residing on a variety of machines. The concept of a "shared" data resource merely implies that all applications have equal access to their necessary databases, regardless of where the information physically resides, and that any new applications installed in the CIS environment will also be able to "share" fully in the existing data resources.

Purpose of The Model

The CIS conceptual model that we have just described is only one of several such models that we might envision. It is not the purpose of this thesis to debate the wisdom of this particular conceptual model, but rather to answer the question: Why is such a conceptual model useful and/or desirable? In the next chapter, we will outline some of the basic advantages implied by the use of such a model, and attempt to summarize these issues to gain a better understanding of how the application of this model can help an organization to increase its strategic advantage.

CHAPTER III

Gaining Strategic Advantage

When we apply a conceptual model of CIS to an organization, we want to better understand and solve the problems facing the organization that result from its disparate computing resources. The fundamental question we wish to answer is: how can the organization utilize its information technology resources to increase strategic advantage?

When we talk of strategic advantage as applicable to information systems, we can talk about either external or internal strategic advantage. External advantage applies to organizations that use information technology as part of or to enhance their product line. Such examples are global electronic banking, software manufacturing, and on-line news and information services. Internal advantage applies within the organization, regardless of the industry it is in. Internal advantage primarily involves improving the quantity and quality of information available to managers within the organization. Internal advantage enables an organization to better use its information resources to improve bottom-line profitability and growth. By improving the quality of information provided to top management, from both a content and cost perspective, organizations can improve their market position and profitability relative to their competitors, thus gaining strategic advantage.

Ways of Gaining Advantage

Irrespective of whether we are trying to gain external advantage as an information supplier, or internal advantage as an information user, or some combination of the two, there are two primary ways in which we can accomplish this task. One way is to gain advantage on the basis of cost, that is become a lower cost provider of information, either externally or within the organization. The second way is to gain advantage on content, in other words have more timely, more accurate, and more complete information than the competition. Therefore, if the CIS conceptual model can either assist in lowering information costs or provide more complete, timely, and/or accurate information, the model can be considered a useful means of increasing the organization's overall strategic advantage.

Although we see that a CIS conceptual model may indeed be useful and/or desirable if it can aid us in lowering costs or improving content, we have still to see just how such a model may accomplish this. It is useful to examine some of the things a CIS model may "bring to the table," in other words, what are the fundamental characteristics of a CIS conceptual model that will provide either cost reductions or information improvements, or both?

Strategic Characteristics of the Model

In a CIS conceptual model of the type proposed by Madnick et al., there are three main strategic characteristics, i.e.

features of the model that directly address the issues of cost reduction and information improvement. They are: (1) Integration; (2) Independence; and (3) Evolution. Integration brings various separate information systems components together where commonality is needed, but at the same time independence helps to preserve the decentralization and autonomy that often exist in larger organizations. Evolution provides for growth and change within the organization, enabling future systems to evolve naturally from the framework of the previously established information structure.

Improving Cost and Content

These three strategic elements offer clear possibilities to both reduce information costs and improve information content. One excellent such example is systems development. In very large organizations, much time can be spent building different systems with similar functional characteristics. Additionally, however, a certain degree of autonomy is often needed in a organization of sufficient size, as it is usually impossible for all the systems development efforts to originate completely from one source. Finally, a great deal of the systems development work in an organization is concerned with changing the existing systems to adapt to evolving business conditions. The CIS conceptual model offers distinct advantages in reducing the costs and improving the results associated with systems development.

For example, the seven-part model has shared data resources and data control to enable commonality between systems to be utilized to the fullest. Fewer separate data storage devices or database machines need to be purchased. Common data needs are shared rather than being developed independently. Data control enables new data to be added to the existing system without requiring new machines to be purchased or extensive time and resources committed.

Since our conceptual model of CIS treats the three separate application layers as being fully independent while surrounded by an integrating framework, organizations can still maintain internal autonomy. Individual subsidiaries or operating groups can develop new applications as the need arises, without being constrained by excessive guidelines or need to conform too closely with other systems. Systems are developed more rapidly, and are better tailored to the business. Manpower costs are lowered, the organization responds more quickly to information systems needs, and the quality and effectiveness of the information systems thus developed are greatly improved.

The same rules apply when existing systems are changed: the CIS model provides a framework for evolution which keeps the application enhancement efforts essentially independent where they need to be, but at the same time allowing them to communicate where commonality can and should exist, to avoid the purchase of unnecessary hardware and the commitment of scarce resources to "reinvent" what already exists.

Many of these same arguments can be applied in reducing

the financial costs and informational inaccuracies associated with human error. Often data is incorrectly or incompletely entered by human operators, wasting system resources in error correction and detection, or worse, compromising or corrupting the organization's information base. Additionally, sometimes top managers lack the information necessary to make intelligent decisions, or have the information but make incorrect decisions. The CIS conceptual model provides for external interfacing and message control to correct and validate data before it is input to the applications, thereby reducing the risk of compromising the shared data resource. In addition, these management and control functions not only aid in providing better information, but they help warn managers of situations in which the applications may be requested to perform functions that may increase risk to the business due to error or miscalculation on the part of the decision-makers within the corporation.

Summary of Strategic Gains

By providing integration, independence, and evolution, our CIS conceptual model enables composite information systems methodologies to be implemented which give rise to cost reductions and information improvements both internally and externally. Though the model we have chosen may certainly not be the only model that provides these strategic gains, it clearly does enable significant control

over hardware costs, software costs, and human resources costs. By providing application independence simultaneously with methods for linkage and communication, it enables managers to tailor information systems to the business, obtain information more rapidly, and get information that is more accurate and free from human error.

In the succeeding chapters, we will attempt to see if the above mentioned strategic gains can be realized when our seven-part model is applied to our "real world" cross-border banking environment. Hopefully, we will gain some valuable insights into the effectiveness of CIS methodologies as a means of gaining strategic advantage in a "live" corporation.

CHAPTER IV

The Case of A Multinational Bank

The subject of our case study is one of the ten largest banks in the world. Since its founding in the early 1800's, it has grown steadily in size as measured by assets, amount of deposits, and number of employees. Its revenues and profits have steadily increased over the years, particularly in the last decade or so. Though the bank is certainly very large even when measured solely on a domestic basis (its branch network in New York alone is among the two or three biggest!) it is even larger when measured internationally. In many regions of the world, including Latin America, Australia, and Africa and the Middle East, the bank is considered to be among the largest players. It is widely perceived to be a dominant force in banking and financial services on a worldwide basis, and thus has numerous opportunities to leverage further global strategic advantage.

Growth and Structure of the Bank

The bank has undergone many structural and organizational changes through the years. In the early days, this particular bank was much like other New York money houses, a place for people to put their money where it would be safe and accessible when needed. During the war years, the bank was a rich source of financing for the mobilization of

America's armies, and later, in the post-war rebuilding era, provided money to rebuild homes and schools, and to get corporations on their feet again.

As regulation and increased competition tightened the commercial banking market in the 50's and 60's, the bank began broadening its customer base in the U.S. and abroad. More local branches were opened to achieve a greater presence in "consumer" banking. A greater effort was made to build franchises overseas, free from the excessive regulations present in the U.S. The lucrative investment banking businesses in New York, London, and Tokyo became much more critical in gaining an edge. The bank was everywhere, steadily building its franchises all over the world, using aggressive marketing techniques and an emphasis on recruiting top talent. The end result by the advent of the 80's was a strong financial services organization, with presences in nearly every major market, committed to being a leader wherever possible, and obtaining strategic advantage on a global basis in every possible market.

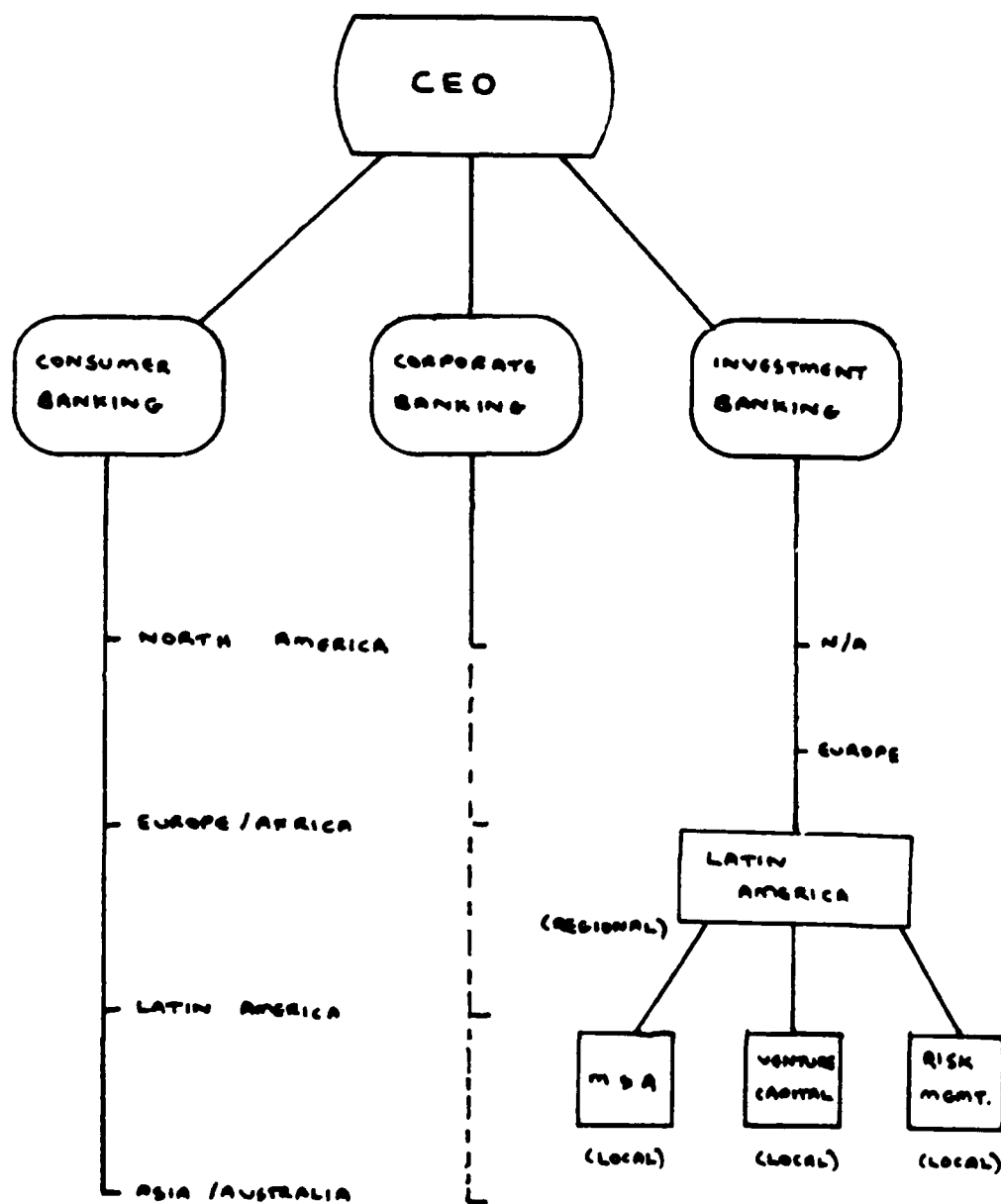
By 1986, the bank had settled into a structure that seemed to provide it with the flexibility needed to maintain its aggressiveness and drive, yet offered the opportunity to coordinate its businesses by market segment and region. The bank is divided into three core sectors: consumer banking, corporate and institutional banking, and investment banking. Each of these sectors is in turn divided geographically along four major regions: North America, Europe/Africa, Latin America, and Asia/Australia. Below this, managers are

grouped either by product line or functionally, with the functional managers generally being based in a "regional" headquarters, and the product line heads residing in the local franchises (See Figure 3).

Each local bank has managerial personnel assigned to oversee activities in one or more of the core sectors. Each "local" sector manager is usually fairly independent from other managers at the same location, e.g. the consumer banking head in Brazil runs his/her operation independently from the corporate banking head in the same country. The structure in general, therefore, reflects both the regional and product-driven nature of the bank's businesses, and gives a sufficient degree of independence to the local managers that they can run the branch banks based on local country conditions, without the need for constant pressure and approval from the top.

The Culture of the Bank

The bank has developed a culture that rewards performance and individual initiative. Managers are encouraged to be very entrepreneurial in their decision-making, and promotions at the bank are often keyed to the degree of aggressiveness and independence that an individual displays. Autonomy and independence are thus encouraged at the local level, and the individual subsidiaries are expected to "run themselves" as much as possible. Local bank heads operate their branches as essentially separate businesses, tailoring



(Parts omitted for greater clarity)

Figure 3. The bank's organizational structure.

their product offerings to what each individual country's consumers and/or institutions desire. Hiring at the local level, particularly for account executive and managerial positions, tends to favor the presence of local nationals, people who understand the markets, speak the languages, and can maintain the relationships necessary to permit the bank to increase its leverage in the region.

Culturally, then, the bank is very aggressive, individualistic, and entrepreneurial. This cultural tenet is a very strong one, shared by virtually everyone in the organization, in all sectors, in all countries, and at all levels. The implications of this culture pervade all additions or changes to the way things are done at the bank, and any new policies or procedures being considered cannot long survive if they substantially conflict with the highly autonomous, entrepreneurial nature of the bank's culture.

Information Systems at the Bank

The pervasive nature of the bank's culture has greatly influenced the way in which information systems planning and development have taken place there. Because of the need for autonomy at all levels of the bank, very rarely has any sort of centralized planning effort been well received or successful. Decentralization has essentially been the watchword throughout. Local branches determine their information systems needs according to the dictates of the local businesses. Functional managers and teams in charge of systems development at a given branch have then

traditionally assumed responsibility for purchasing that branch's hardware, software, developing functional requirements, customizing, and documentation.

With each individual branch and/or sector being responsible for their own information systems policy, coordination of these information systems into any kind of coherent framework has been difficult at best. In the initial stages of systems development and implementation at the bank, this "shotgun" style approach, with everybody doing their own thing, was not so bad. In fact, it actually speeded up the process of getting certain types of functions to work well. This was so because many different groups might more often than not be working on the same problem simultaneously. With several groups working on the same problem instead of just one, often the solution that the bank needed was arrived at much more rapidly than if a centralized effort had been attempted.

A major drawback of this approach, however, was that it could more often than not be very costly and waste precious resources. As banking was becoming more competitive on a global basis, with many "regional" and smaller local banks putting pressure on the bank in many countries, the costs of a decentralized approach to information systems become much more apparent. Additionally, bank management was beginning to realize the great strategic potential that lay in global electronic markets and securities, communications networks, and technology-based global financial services. As the bank

became more and more committed to penetrating and holding an edge in these kinds of markets, it became clear that more integrated, "global processing" systems were needed. These are essentially systems that provide some commonality across different branches, different sectors, and different countries. In other words, the bank had to attempt to bring at least some degree of standardization and integration into its information systems structure, so that communication between countries and across different markets could be achieved in a more cost and resource-effective manner.

The Introduction of "Global Processing"

As a result of the notion that at least some attempt at centralizing the information systems process would be a good idea if the bank was to continue to stay competitive, attempts were made in various groups of the bank to consider the "globalization" potential of a number of existing or planned systems. The idea was to try to find some functional or data needs that were essentially common to many areas of the bank, and try to impose the systems handling these processes onto multiple geographic locations and/or sectors. It was hoped that such a strategy would not only reduce the higher development costs brought about by redundancy, but also would improve the feasibility of communication between the different systems in place around the world, and better enable the bank to attain strategic advantage in globalized markets.

Some of the systems developed during this period can

rightfully be called "global processing" systems. These are not necessarily Composite Information Systems as defined earlier, but they do have the capability of handling certain kinds of applications on a globalized basis, largely due to the fact that theoretically, the applications being handled do not vary much from region to region. For example, most general ledger and accounting applications are fairly universal in their basic structure and functionality, although slight implementational differences may exist such as different currencies, taxes, etc. For our purposes, we will define a "global processing" system here as a set of computer systems and/or applications that at least have some elements of globality, that can be coordinated across countries and perhaps even sectors to provide some overall worldwide integration.

We propose to describe in the next chapter a certain existing type of global processing system, referred to here for convenience as GPS. We will then examine the nature of GPS as it exists today, with the long-term goal of determining how a CIS conceptual viewpoint, as opposed to merely a "global processing" orientation, can impact upon the strategic advantage that the bank can derive from the use of GPS.

CHAPTER V

GPS: A Brief History

Of the many computer systems in existence at the bank both in the past and today, GPS has been and is one of the most important to the bank's success. Historically, GPS began as a stand-alone system for a single country. Through evolution, it became an attempt to impose some commonality on the already widely decentralized and separate information systems components of the bank. The theory behind this gradual migration of GPS was straightforward: attempt to find common elements in various autonomous divisions of the bank that varied very little from country to country, or from application to application. Although GPS was not developed to be a "Composite Information System" as we have thus far defined the term here, it nevertheless was and is a significant system from the bank's perspective, both because it was one of the first true attempts at commonality within the bank, and although it has met with many failures over the years, it also has achieved its share of major successes.

The Origins of GPS

GPS was originally developed around 15 years ago, in the early 1970's. The original system was European, and was designed to service European clients. As we previously noted, GPS was not originally designed to function across

borders, rather in one specific country. However, when the system was first installed, it worked so well that many other local European managers wanted the system for their branches. The system proliferated steadily throughout Europe, spurred on through the enthusiasm of local management, and the response was so positive that in 1976, bank management decided to implement GPS on a global basis.

There were many clear reasons why a global migration of GPS was desired by bank managers, other than the fact that it worked well in Europe. Chief among them was the fact that the bank had little, if any, viable processing systems that could handle a wide variety of transactions at any level, never mind a global basis. Most of the systems that existed to process transactions at the local level were stand-alone batch systems, highly antiquated, and generally not capable of handling large volumes or types of transactions that deviated from the "status quo." When faced with increasing competition, the bank's obsolete batch systems could not provide a level of customer service and support commensurate with the rest of the industry. The bank was losing competitive advantage in the marketplace. In the words of one manager, "Something had to be done."

Thus the bank had to act. They needed to have something on the table if they wanted to continue to stay competitive. If for no other reason, GPS is significant in the history of this specific bank in that it marked one of the first times that information systems were developed as a strategic response to the competition. GPS, in effect, was a "training

ground" for the further use of information technology as a powerful strategic weapon within the organization.

Another reason for the implementation of GPS centered around the emergence of technology-based financial services and electronic banking as increasingly important markets for the industry. Consumer and corporate customers alike were eager to have - and pay for - electronic funds transfer, automatic teller networks, and cash management services. Many of the bank's competitors were entering aggressively into these markets, and the bank felt an acute need to respond competitively to the challenges imposed by the new "high-tech" slant of the banking industry. The above issues of needing to react strategically and to provide more "high-tech" services gained further urgency as the bank grew in size, as transaction volume was rapidly increasing. Without more sophisticated internal computing systems in place, that were better integrated and less costly to operate, it would steadily become even more difficult to offer truly competitive products and services.

A third reason for change was that the banking industry was shifting more to fee-based services, as these offered more profit potential than the dwindling interest "spreads" which were the traditional "staples" of the bank's income. To achieve strategic advantage in fee-based markets, banks had to have faster and more accurate information processing abilities than their competitors. This was especially true due to the fact that the most lucrative opportunities for

fee-based services were generally considered to be technology-intensive applications, such as cash management, global securities, and electronic funds transfer. And even the less technology-focused applications were gradually requiring more automation and less human intervention to be efficient. In general, information systems were becoming more of a strategic weapon than they had been in the past, and the bank's top management saw GPS and systems like it as first steps towards a more integrated and efficient information systems structure.

The Purpose and Function of GPS

GPS as originally conceived had bold ambitions: to provide real-time processing functions for many of the bank's major products. These included traditional products such as loans, demand deposit accounting, letters of credit, etc. and could in time include electronic funds transfer, cash management, and other more modern "technobanking" products. As GPS gradually became more "global", its main purpose evolved to providing faster, more efficient processing of transactions around the world, offering some degree of functional and data commonality between countries, and assisting in integrating the bank's global operations to enable the organization to increase its global strategic advantage.

Bank management believed that if every country had a version of GPS, it would be a fairly easy task to "standardize" global processing operations around the world

to at least some degree. This would have the twofold advantage of reducing development costs and permitting the bank to respond more rapidly to competitive pressures. Most importantly, the bank's management viewed GPS as a step towards integrating more completely the different countries and sectors in which the bank provided services, by providing some elements of commonality that all countries and functional areas could depend on having. In this way, as the bank became more global in its orientation, it could better serve customers with truly "global" banking needs. For example, GPS could eventually enable multinational customers to consolidate funds spread out in many different countries or branches, or assist branch banks in giving loans by providing lines of credit drawn on funds existing in more than one country or geographic region.

Technical Characteristics of GPS

In considering the effectiveness of GPS as a "global processing" system, or even as a Composite Information System, it is important to note that GPS was designed and built with the technology of the 1970's, much of which would be considered obsolete by today's standards. For example, GPS was written entirely in COBOL, a language well suited to batch processing, but perhaps less well-suited to other types of processing. Development was based around a series of "modules", one for each functional application required. The theory was that some branch banks might need all the

modules, some might need only certain ones that were appropriate to their particular applications. All modification would be done at the local level, only an "off-the-shelf" version would be delivered from the European source.

No database management software existed in GPS as originally written. Summary reporting was arrived at through meticulously "tagging" and saving individual transactions, and sifting through them at the proper time to generate reports. Thus we see that GPS originally was a very storage-intensive system, with high system overhead required in processing the large numbers of transactions involved. There were no shared data resources as such, and control of data and messages was nearly impossible to achieve. Indeed, as the numbers of transactions processed grew steadily larger, the tedious counting and summarizing of records could potentially cause huge response-time bottlenecks.

Thus, any commonality that existed as a result of GPS could really only add value from a functional perspective. Where GPS was perceived to really provide benefits was through reducing the time and money spent in developing individual applications. By creating standard "modules" that all branch banks theoretically could use with only minor modifications, the bank hoped that systems development efforts at the local level in response to changing needs could be greatly simplified. However, as we shall see, the effectiveness of this approach at an individual branch depended largely on the way in which local management

worked with top corporate management to help implement GPS, and how useful GPS was in addressing that branch's specific applications needs. As a result, though some branches of the bank considered and still consider GPS to be a success, there are just as many others that viewed and/or continue to view GPS as either a partial or a complete failure in a number of ways.

Implementation of GPS: Failures and Successes

When the decision was first made by bank management to implement GPS on a global basis, the intent, as previously noted, was to have each local branch install what functions of the system they required, and then customize at the local level where appropriate. The key to how effective this approach would be in part depended on how much "customizing" would really be required in a branch bank, i.e. how much of the system was actually common "across the board." Although originally it was thought that nearly 100% commonality existed, as implementation began it was clear that a good deal less was actually present. The more successful implementations of the system within the bank tended to have needs that more closely corresponded with the exact needs for which GPS was originally intended. Indeed, some managers complained that GPS was "too European" [4] and would require extensive changes just to make it remotely feasible in branches that had very "non-European" business requirements and needs.

In cases where local modification was in fact required, it was extremely important to have people on hand who understood COBOL programming, knew the software techniques necessary to make changes, and who were thoroughly familiar with the functions of GPS and how they were programmed. This often presented problems at local branches where language differences existed, because much of the documentation was cumbersome and often very difficult to translate. In addition, since the bank had traditionally hired mainly local nationals into the branch banks at most levels, difficulties often arose in lesser developed countries, where qualified software experts were scarce. Therefore, the bank's initial assessment of the relative ease of local modifications often turned out to be very different in actual practice.

Another major difficulty in implementation grew out of local management's traditional independence and autonomy in their operations. Though all the bank's branches have a certain degree of autonomy, as we have noted, some branches were naturally more independent than others. Sometimes this was so because country conditions dictated it, other times it was simply due to the personalities of the managers involved. Whatever the reasons for this independence, it was considerably more difficult to introduce a centrally planned system such as GPS into the more autonomous units of the bank. As we observed previously, central planning has seldom if ever been an internal virtue at the bank, and GPS was no exception. Top management had to foster a strong

relationship with the local managers and convince them of the feasibility of GPS before many of the local managers would take a chance on installing the system and committing developmental resources within their respective branch banks.

The final issue to note surrounding the historical development of GPS at the bank has to do with the extreme variability of control over the functions of the system. While in Europe, where GPS was developed, the system was designed to enable all transactions to be carefully audited and monitored, this process would not necessarily serve in other countries, where local auditing and control requirements were very different. With an integrated, global processing system such as GPS, security and control requirements were of necessity more complex than in a more traditional batch system. The lack of effective controls posed little problem in some branches, but in others it was positively disastrous. Some of the competitive advantage in cost reductions and information improvements that the bank had hoped to realize as a result of GPS evaporated in some branches due to the inability to control and monitor data. As a result of these inconsistencies, the bank had some branches that saw GPS as stable and effective, and others that viewed it as an absolute disaster, and a serious liability that hindered maintaining strategic advantage in local markets.

Yet despite the problems that existed with GPS as

originally installed at the bank, many managers considered it to be a success. GPS did provide a good deal of commonality and increased integration for certain branches running certain kinds of applications. And clearly GPS gave bank managers a sense of which data and functional linkages were really important, here commonality really mattered. GPS was successful initially in providing faster response times and more efficient customer service. It did help to reduce the costs of transaction processing. Another important point is that GPS greatly simplified user interaction with a number of common functions, such as loans, letters of credit, funds transfer, etc. and did offer a more generalized reporting structure so that reports to Corporate could be more easily created and maintained.

The last factor worth noting is that GPS, despite its weaknesses, was still a viable, functioning system. GPS still was light years beyond anything that the bank had possessed up to that point, and took full advantage of the technology available at the time. For many branches, it was enough simply to have a system that worked. There was a certain comfort in this fact more often than not. Indeed, many managers felt that GPS was a "stable system" that provided well for their needs, and even today some of these same managers swear by GPS, and are very reluctant to let it go even if a potentially superior system could be found.

Summary - Where GPS Stood

In short, when GPS was first implemented on a global

basis at the bank, the reviews were highly mixed. Some branches found that GPS helped their business considerably. Others were not so convinced. And some branches were positively livid. But in general, the bank lived with the system, and continued to thrive and prosper. Over the years, many changes were made to GPS, both on a local basis and globally. Now, in 1987, GPS still exists at the bank, but it is quite different in a number of ways. Some of the old problems still exist, but some have been cured, and new ones have taken their place. Some of these changes have stemmed from technological reasons, others from organizational concerns. And there are a number of ways in which GPS has not responded to change, partly because it is too outdated to deal with the change technologically, partly because the bank's culture does not lend itself well to implementing certain kinds of change.

So what does GPS look like now? And how might a CIS conceptual viewpoint make things different? We shall attempt to answer these questions in upcoming chapters.

CHAPTER VI

GPS Today: The Process of Evolution

Before we examine how GPS is today, it is extremely important to note how GPS has been influenced by technically-driven and organizationally-driven changes that have occurred at the bank in the past 15 years. This stage of the analysis serves two main functions: (1) it will help us to determine exactly how, and more importantly, why GPS did not work in the past, and/or does not work now; and (2) we will get a better sense of what things might be done to correct these problems, and how specifically a conceptual CIS viewpoint may or may not assist in these corrective measures.

To get a better idea of how GPS has evolved, and of the changes that have brought it to where it is today, a series of field interviews were conducted at a number of sites in the U.S. and abroad. Many of the opinions expressed in this chapter and in subsequent chapters, therefore, often reflect the views of local managers in a given country, and do not necessarily reflect an overall "universal" opinion regarding GPS. The most important issue to be aware of is that there is no universal opinion within the bank. Every local manager has a different view of the changes that have taken place from his or her perspective, both technically and organizationally. Consequently, the attitude towards GPS, its evolution, and overall effectiveness at a given branch

can vary widely within the organization, depending on the particular circumstances involved.

Basic Issues Surrounding GPS

When GPS was first installed, as we have seen, its effectiveness varied considerably from country to country. Branches that required applications very similar to those for which GPS was designed, or that were less "ruggedly individualistic," as it were, fared better. This by itself perhaps could have been overcome if the bank had been prepared to deal with these difficulties, but as it was, the amount of modification and guidance needed at the local level was severely underestimated from the beginning. The bank had assumed from the start that the "core" GPS system would require only minor modifications to be effective in all countries. Given this viewpoint, they neglected to set any clear policy guidelines for change, or to provide detailed and accurate system documentation to the countries. As a result, countries that needed to change very little did OK, but countries that had completely different needs from those supplied by GPS had troubles, both with understanding what the system was about and the proper procedures for going about changing GPS to suit their needs.

Since documentation was extremely poor, most countries had to rely on a "hit or miss" approach to make alterations to the GPS system. As one manager in Latin America said, "We had millions of lines of code, whereas we only understood the real purpose of a few thousand of those lines. We had to

improvise." In short, many countries forgot about trying to modify GPS as supplied and wrote their own software from scratch, "tacking on" various stand-alone programs to the GPS databases. More often than not, such an approach didn't work on the first try. "It's very wasteful," agreed one technical head, "but we don't have any choice. We have to run the business as we see fit, despite GPS."

Therefore, a good deal of time was spent in some countries writing and rewriting software as GPS evolved. Not only was this expensive, it made it even more difficult to follow what policies did exist for maintaining GPS on a "centralized" basis. For some branches, even this question was moot. "'Central' maintenance is a joke when you're talking about GPS," said the technical head. "We can't do anything that other branches do, because we don't even have the same hardware. Out of the original GPS modules we received in the 70's, we've only retained one or two today. There is no question that we're completely cut off here from any possible 'global' changes to GPS ... even the last memo I received on the subject from anyone was about a year and a half ago."

The interesting thing about the evolution of GPS in this case is that initially, the bank had at least attempted to implement some global mechanisms for controlling and monitoring modifications. In the initial stages, in fact, no modifications to GPS were allowed at the local level without specific procedures being followed. However, as local banks

started pursuing business more aggressively, this requirement was largely ignored, and soon fell by the wayside. Today, it is widely recognized, even at the corporate level, that trying to impose any sort of structure on the GPS development process is usually a waste of time.

"You have to understand what we faced here," said one local bank head. "Here is our little branch bank, growing at an exponential clip - very overworked and understaffed - and people at the top saying we can't change our system unless we go through the whole approval procedure. First of all, it takes too long, we need to respond now. Second of all, it ties up precious manpower and development resources that we just don't have. Third, they just don't understand our needs. I mean, in the past, nobody from the GPS development group even came down to visit our operation, and here they were, several thousand miles away, telling us to 'hold everything', when we had an important crisis or opportunity that we had to respond to on the spot. I'm not saying they wouldn't be sensitive to our problems if they really understood them ... but they don't understand them, so they (the GPS people in Europe and New York) really can't be of much help to us. GPS is our own baby now, and I suspect it will remain so, at least in this country, for some time to come."

What we see as a prevalent problem during the evolution of GPS is that top management often just wasn't aware of the extent of change required and the need to respond to it at the local level. The lines of communication were often very

poor, and became worse as the bank grew. Although GPS ostensibly began as an attempt at commonality, throughout the years this commonality became diluted at many branches as the "local" GPS became more and more cut off from the "central" GPS. What this meant is that GPS evolved from being one system into being many different systems. It became a common joke around the bank to refer to GPS as the "Heinz system" because there were at least 57 varieties of GPS floating about in the various local branches.

Coping with Local "Isolation"

Because of the gradual transformation of GPS into a greatly varying set of multiple systems, what fragile communication links did exist between central GPS management and local GPS management were severely tested. Even branches that were fairly satisfied with GPS felt the strain. Another Latin American branch head admitted that very few changes had been made to GPS and that GPS was "fairly stable" at his branch. But the tenuous communication with central GPS management still caused problems. "There still is no documentation, and I don't feel comfortable going to the central group for help, because I don't get a quick enough response," noted the branch head. "Not only that, we never know which version of the operating system they're using when they send us a change, so sometimes we load it up and it just crashes."

"Crashing" is a problem that has partly evolved out of

the isolation of the branch banks from the U.S. and Europe. In lesser-developed regions of the world such as Latin America, it is difficult to employ sophisticated hardware and obtain competent technical expertise to do the complex maintenance that GPS more often than not requires. Sometimes branch banks purchased whatever equipment was handy or could be imported at reasonable cost, but never had enough support from outside to understand the technical feasibility of what they were implementing. Therefore, often systems "went down" without notice simply because the technology was not fully understood or well utilized.

"Size and quality of computer equipment are definitely critical success factors from our standpoint," noted one member of a local GPS maintenance team. "We crash a lot, we'd like to avoid it, but our machine is too small for our applications volume, and is frequently overloaded. We also have lousy communications equipment, the phone service here (in Latin America) is generally very bad ... We don't get much support from the people in the States on this, it's hard because everyone is overworked. We also have different hardware from most other groups, and we have too many 'standalone' kluges. What I'm basically trying to say is that we are so completely different from the rest of the bank that we are the only ones who can fix things. Nobody else really is in a position to understand our equipment and setup ... It would take an outsider several months just to sort through all the changes we've made in the past 15 years and figure out, generally, just where we stand."

This same group member also admitted that a large part of the problem has to do with being located in a lesser-developed region, which makes solutions doubly difficult given the bank's culture. "Good technical experts are hard to find in this country, or indeed in most of Latin America. The bank traditionally has tried to hire local nationals into key positions, and I laud that. But we're often stuck between a rock and hard place because of this strategy. We are fairly isolated, as I said, so we are the only ones who can solve our technical problems. But we suffer from a shortage of technical talent, and it's harder to train people from within. If we were in Europe, perhaps, we might not have this problem, only because a lot of technology transfer has already taken place over there, and it is a lot easier to find people who can write in CICS, for example, or perform some other critical technical function that we need."

Critical Success Factors: Did GPS Help Achieve Them?

Each country in which GPS was initially implemented had in mind several critical success factors which were essential to the sustained growth of the local business and the increase of local strategic advantage. As should probably be obvious from what we have seen so far, each branch had different critical success factors that often applied solely to it, and had no bearing whatsoever on other countries' needs. However, we can make some general

observations regarding the critical factors that GPS directly influenced, and how they have evolved over time as GPS gradually reached the point it is at today.

To begin with, the primary critical success factor in most branches was one of the main issues leading to the development of GPS in the first place: provide fast, efficient, and accurate services to customers. With the old batch systems that existed in most branches prior to the introduction of GPS, it was very hard to achieve this goal. Corporate and individual customers alike required speed and accuracy of transactions, which most of the "pre-GPS" systems could not handle well due to the volume. Many branches processed more than 100,000 transactions daily. These could not be processed swiftly and efficiently without a system such as GPS.

Another critical success factor nearly as important as the quality of services was the range of services available. Customers had become fairly sophisticated and demanding in their requirements even in GPS's early years, and wanted to take their businesses to banks that offered everything - somewhat like "one-stop" shopping. One way in which GPS initially helped this problem was the modular format, which enabled new functions to be added to an existing GPS configuration in a very short time. The obvious problem with this approach, however, was that only a finite number of modules existed. Thus, as GPS evolved, and the countries in which it was installed came up with new applications, eventually countries began to "run out" of available GPS

modules, often for extremely critical applications from their point of view.

"It's pretty unbelievable when you think about it, but our original version of GPS couldn't process checking accounts," noted one operations manager. "Imagine, a fairly 'routine' application from our point of view, involving virtually all the individual customers and many corporate customers, and GPS couldn't process checks! This was a classic case of the guys who built the system not being in touch with the global environment of the bank ... In Europe, where the system was built, virtually nobody had checking accounts with the bank, mainly because the corporations and financial institutions there were essentially 'checkless', using funds transfer, drafts, and so on to move money around. The system wasn't designed for retail banking as such, so checks weren't really used in the 'original' version of GPS at all."

"However, when the folks in Europe had to hand off this system to us and others like us, they didn't do anything about the check issue ... They may have thought about it, but they didn't do anything. They didn't realize we have a lot of consumer banking here, they looked at the system and how it was, and naively assumed that to be the case all over the world. Thus we had to spend a great deal of time writing a new module, which was very difficult to do considering the lack of good documentation and central support ... but I'm sure this problem isn't unique to us. There are several

other branches who also process a lot of checks."

As GPS evolved, this need to provide better and more varied services led managers, as was previously noted, to improvise. Without a clear methodology for responding to change, and without an ingrained understanding of how changes would affect the overall system, bank personnel spent a large part of their time adding interfaces haphazardly, just to get things to work. Thus, the user interfaces between various parts of the system gradually became more cloudy and difficult to control.

"GPS is sometimes very confusing to me," noted one account executive. "I have 5 different terminals and at least 15 different manuals telling me how I'm supposed to access the system, but I'm often not sure exactly how everything fits together. When a customer calls me with a problem, that some transaction hasn't gone through, or their statement is wrong, sometimes I feel it's a matter of luck if I can find the solution to their problem. After working with the system for a while now, I sort of have a 'method' in my head for doing everything, but I still don't understand it. GPS is certainly not very user-friendly - God help me if anything should change, it would take me weeks to learn everything over again!"

Because of the breakdown of the user interfaces in many instances, another critical success factor that has evolved is to make things more user-friendly. This is considered to be an extremely monumental task in most cases, but is viewed as essential if the bank is going to be able to stay

competitive. "Hey, GPS was originally put in here to make it easier for us to do our jobs," noted one product manager. "I agree that it has helped, but a lot of us still aren't very comfortable with the computer ... Often we just do things by hand rather than 'face' GPS, and that isn't very productive. The problem seems to me to be that the system has gotten so big that it's like a monster ready to pounce, everyone's afraid to touch it for fear that it will bite their hand off."

The final critical success factor that seems to be pervasive throughout the bank is the need to respond rapidly and efficiently at the local level to changes in banking regulations. This is the factor to which GPS has traditionally been least responsive. Unfortunately, it is increasingly becoming the most critical factor in many branch banks' operations. "GPS was great 15 years ago," stated a local technology head. "We had a great deal less transactions than we do now, but even more importantly, our needs didn't change all that much. Now we've had some important governmental policy changes (in this country) in the past few years, changes that cause banking regulations to be different almost every day. GPS can't handle these new requirements, it's too outdated, and the code has increased to the point where it's almost unmanageable. There are some modules of GPS that we can't even use any more, because we're unable to alter the code fast enough to keep pace with local regulations."

AD-A193 856

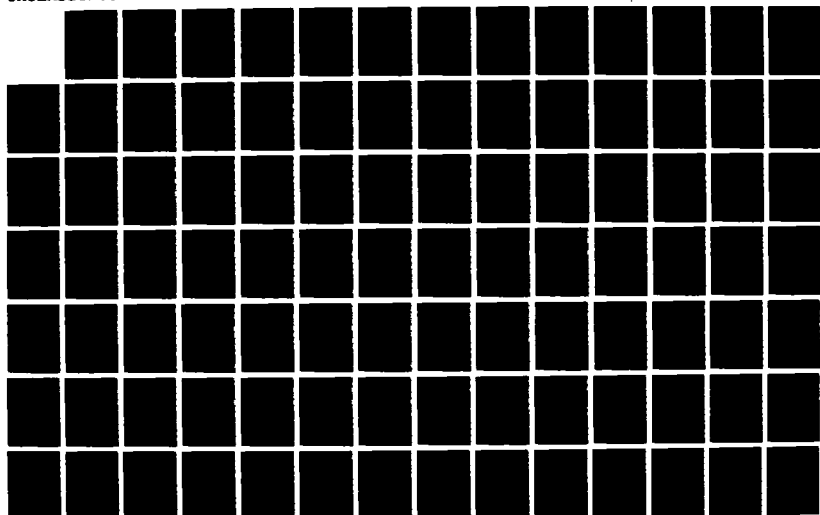
INTEGRATING INFORMATION SYSTEMS IN A MAJOR
DECENTRALIZED INTERNATIONAL OR. (U) MASSACHUSETTS INST
OF TECH CAMBRIDGE A GUPTA ET AL. DEC 87 MIT-KBIISE-7
F/G 12/7

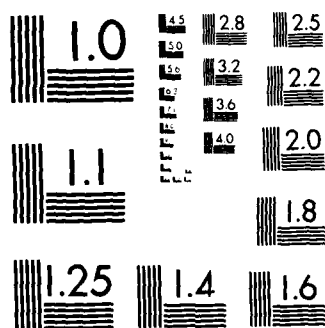
2/3

UNCLASSIFIED

DTR557-85-C-00003

ML





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

"For example, I have 10 boxes of computer printouts containing old GPS code which we don't, can't use anymore. It's often better simply to spend the money and write a new system that duplicates what we already have, because we'll at least come out this way with a system that we understand and can use. Take loans for example, which is a fairly common and critical application for us. We don't use GPS to process loans any more, because frankly, it's too damn complicated. It's too much of a hassle. We've written our own loan system which sits on the PC ... Does it hurt integration? You bet it does, but that's tough. We can't run our business aggressively if our computer systems are always lagging well behind our needs."

Where GPS Stands Today - Are There Solutions?

Today, GPS has evolved into something very different from what it originally was, or indeed what it was originally intended to be. Much of the original commonality that was a main thrust of the system has all but disappeared. Though bank managers all over the world disagree on the extent of the problems surrounding GPS, one fact is generally agreed to be true: GPS runs. "In one respect, GPS has accomplished at least part of its original purpose, and that is that it does work," noted a regional bank head. "It's a ghastly process, making it work, it's very costly, it ties up a lot of resources, but it can be made to run in most cases. The worries I have aren't so much about what's happening now, but what are we going to do in the future? We're spending

too much money on keeping it running, and we don't know whether our efforts will be sufficient to get GPS to stay running in the future. I firmly believe that we need to think sometime down the road about replacing GPS with some other system, but at the same time we have to keep GPS in good working order until we replace it ... Despite all that's gone wrong with GPS, we're too dependent on it now to just drop it. We have no choice but to grit our teeth and stay with it."

The above comment sums up well the underlying opinion of many bank managers: GPS may not be so great any more, but it does work, and the bank is fairly dependent on it. However, though many agree sticking with GPS is the only option now, they also concur that something new is needed down the road. The question is: what? Almost nobody is in agreement on this, but they do have some insights into what some of the main problems are, and what possible solutions might exist to solve these problems in the future.

"I think one thing that is clear from looking at GPS today is that top management underestimated the problem that we would have 'in the trenches'," noted a systems manager. "There wasn't enough commitment to really working with the local management to understand their specific requirements. GPS was just thrown at us, without giving us a good idea of what was expected, what it could do for us, how it would address our needs. I think the most pervasive flaw in GPS is that the people who rolled it out were too far removed from

what was going on in the countries. It shows too, because what we have is a hodgepodge that only does the job with considerable investments of time and resources. Communication between management groups could definitely have been improved."

Other managers agree, and take it a step farther by pointing out that much of the problems with GPS stemmed from the bank's own aggressive internal culture. "There is no doubt that communication was bad," pointed out one manager, "but you have to understand that lack of communication was and often still is a very pervasive feature of the culture of this organization. We're a very aggressive bank internally, almost too aggressive at times. Everyone wants his or her own piece of the pie, and doesn't want to share it with anyone. People who produce, even at the expense of others, are rewarded here, while people who are always being helpful to someone else at the expense of getting their own work done really suffer ... It's not a culture that always fosters communication, or even sensitivity. Everyone around here really has to fend for themselves."

Given that there are many organizational obstacles towards arriving at effective solutions to problems within the bank, many individuals admit that technical solutions may be hampered in effectiveness as well. There is a fair amount of consensus which says that the best technical solutions should consist of smaller systems that are well documented, user-friendly, and are less difficult to modify quickly. Many managers also believe that 'commonality' isn't

really an issue. What really matters, in their view, is that they don't lose their ability to respond to business conditions and to make changes as they see fit. Thus, many of the bank's managers really want to achieve independence and evolution, perhaps even at the expense of integration.

So how can we sum up the problems? On the technical side, we have an outdated technology, poor documentation, the unavailability of certain applications, slow response time, poor user interface, difficulty of control, and overall lack of flexibility. We also have a hitherto overlooked but nevertheless critical point: a lack of design. Noted one outside GPS consultant, "GPS was never really planned the way it should have been ... there was never any design methodology, or attention to the overall picture. Nobody even knew originally what hardware GPS should run on ... the overall poor planning has caused this bank numerous problems in both huge amounts of money being wasted and people's time being incorrectly utilized."

On the organizational side, there was no communication with users, no formal GPS policies were established, and the aggressive, autonomous culture of the bank often greatly inhibited management groups from working together. People at the bank generally wanted to do things their own way. That was why such people were often hired in the first place, but it also helped to foster an attitude that made commonality, or even integration, of GPS a very difficult task.

Are there any good solutions to the problems that have

been suggested thus far, that are both technically and organizationally feasible within the bank? Perhaps, when we attempt to redefine the GPS environment using our CIS conceptual viewpoint, we will be able to answer this question a little better in subsequent chapters.

CHAPTER VII

Application of the CIS Model

Now that we have explored the evolution of GPS from its early origins to where it stands today, and gotten a better feel for the nature of the problems and inconsistencies posed by GPS at the bank, we can "redesign" the existing GPS system from a CIS conceptual viewpoint. Our intent here is not to propose that GPS is necessarily "right" or "wrong," rather the idea is to show how a CIS conceptual view of the system might: (1) improve some of the problems we already know exist with GPS, (2) make some of these same existing problems worse rather than better; (3) perhaps create new problems that do not already exist with GPS; and (4) somehow improve global opportunity, i.e. allow the bank to increase overall strategic advantage.

When we consider the ways in which our CIS conceptual model may be applied to the GPS environment, we must note that it may not always be easy to prove these results through direct observation. Some elements of a Composite Information System as we have defined it do in fact exist now at the Bank, but some solutions we propose may only be intuitively obvious, and may not be feasible just now, either because the required technology cannot be implemented cost-effectively, and/or the strong culture of the bank precludes making such a change. Even if such solutions are in fact "feasible" from both technical and organizational

standpoints, we are only making observations, we are not recommending anything. It is not the purpose of this thesis to tell anyone how to run their business, but rather to point out some features of a CIS environment, both in terms of the problems they may pose and the solutions they may generate.

Where a CIS Viewpoint May Help GPS

From the preceding chapters, it is clear that "redesigning" the existing GPS system from a CIS perspective will solve at least one problem: the lack of a design. "If we had to do things all over again, there's no question that a good design would be number one on our list," notes the outside GPS consultant. "We need to have design teams that understand the user needs. I'm talking about locally based teams as well as centralized ones. There should be a couple of people at each branch who understand the applications very well, and can work with the managers at the local level to solve their specific problems."

"When GPS was built, the technology was very primitive, I don't dispute other people's claims there. We didn't use relational databases, we had no data modeling, no fourth-generation machines, nothing. But lack of technology wasn't the long-run problem. Lack of design was. It made the past an anchor which held back all future development. We failed to plan for the future. We weren't flexible."

Flexibility. That has certainly been a key word throughout this research. It is quite apparent that

flexibility in many cases is the ultimate golden ring at the bank, everybody strives for it, but never quite achieves it. Can a CIS conceptual model help with flexibility? In many ways, yes. The primary advantage of the CIS conceptual model in promoting flexibility over a more "traditional" systems development approach such as GPS initially used is that the CIS conceptual model specifically recognizes the need for independent applications. The integration occurs in the inputs to and outputs from the applications, not necessarily in the applications themselves. Our CIS conceptual model doesn't rely on a completely vertically integrated set of applications as does GPS. Rather, the actual application functions are developed independently, while the commonality lies in the transfer of information to and from these applications. This may be better illustrated with a simple example.

Let us suppose that we want to develop a very simple system in our bank that only has one module, to process checking accounts. Let us furthermore assume, as is currently the case with GPS, that we want to implement this system in many different countries without knowing the rules, i.e. we do not yet know what local modifications will be required in each specific country. We do want to get the system out the door, though, so in the initial release we will design our application around a "base case", usually representative of the way checking is done in our country. Thus if we are starting off in France, we will have a

"French" checking account system, which may or may not be the same as a checking account system in Spain, Germany, Japan, or Brazil.

The GPS approach says: Let's develop the whole application top to bottom, so that we get it working perfectly in our region. Let's "hard code" all the input screens and output reports up front, and put the majority of the burden on the countries to modify it locally. When we add, say, a savings account module to the system later, we will develop it exactly according to our base case, and treat it as a completely separate module from our checking account system. However, we'll try to develop a certain "core" code as we go through the process for similar applications so that each new application takes less and less time to build. Our commonality will be across functions, we'll try to "link" our applications together with common code. So for example, once we develop a subroutine to print our checking statement, we can use it for the savings account module as well.

The CIS approach says: Let's not worry about "core code" or the real substance of the applications, let's instead ask: What are the common informational needs here? Let's try to start by looking at critical success factors in different branches and try to understand data needs. Let's try to start by building a common database. Let's try to develop common input screens and interfaces with users, and only do so once. Similarly, let's try to define the rules for storing and sharing data in some way that seems to fit

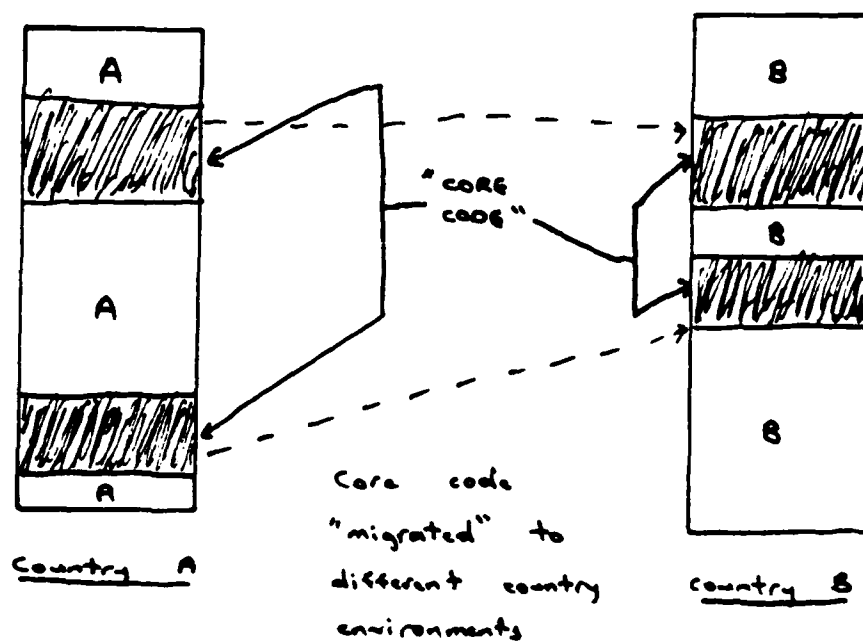
across countries. It may take more effort on our end to do this, but the local countries can develop whatever applications they want, and "fit" them into the integrating framework we've established. With our approach, we aren't looking to develop a checking account statement subroutine so much as we want to come up with a checking account database (See Figure 4).

Although this is a simple example, it is designed to make the point that the CIS conceptual viewpoint recognizes that applications can and should be independent, and provides for this independence as well as future evolution, with integration of the data resources and linkages with the outside world as the primary means of achieving commonality.

In what other ways would a CIS conceptual perspective add value to GPS? Well, the idea of a consistent user interface and better data and message control is particularly appealing to some managers. "It would be so nice to reduce the number of errors we have in GPS transactions," commented one operations manager. "One of the good things about GPS is that it has greatly reduced the quantity of work we have to do manually. Of course, some human intervention is inevitable with any system, you can't avoid it entirely. But people don't have a lot of confidence in GPS in some cases, because a lot of data slips through the cracks. I think it would be a big step forward to have a stronger link with users, a feeling of confidence that the data we're giving to customers and running our bank with is really correct."

GPS

"Checking Module"
(subroutine)

CIS

"Checking Database"

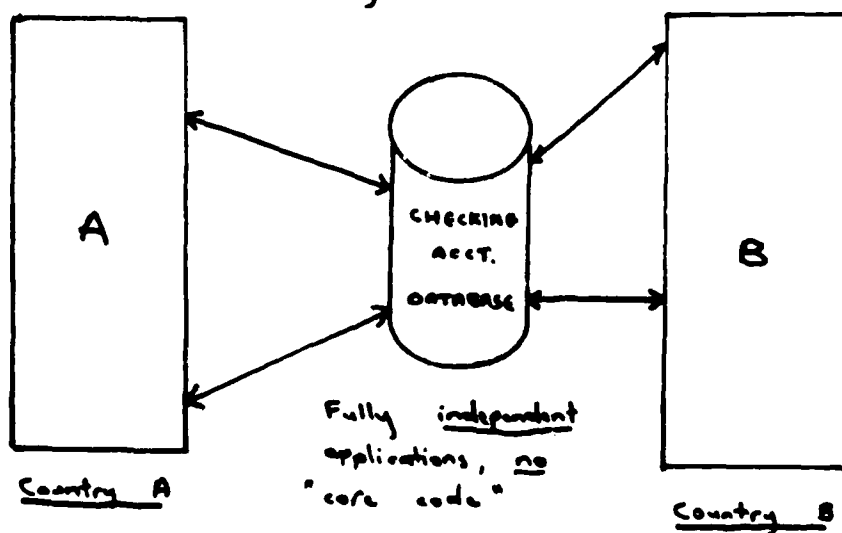


Figure 4. GPS vs. CIS approach to systems development.

"Of course, there's no way to be certain that everything is 100% right, 100% of the time. But if we could just give users a better sense for what kinds of data are really valid, and have a system sophisticated enough to help them with this task, well, then I think we'd be substantially ahead of the game."

There is no doubt that a CIS conceptual model can help in other areas that perhaps are not serious liabilities of the GPS system, but are important nevertheless. One such area is that of risk management. As we have previously noted, risk management can help determine whether "reasonable" limits of the data are being violated, such as: Does the bank have less cash on hand than reserve ratios require? Another important function that the CIS conceptual model provides is the notion of linkages between the transaction-processing functions of the business and the information-processing, analysis-intensive functions, a task that simply defies solution by GPS.

"GPS has been traditionally thought of as a transaction processing system, and that's basically what it is," observed a local GPS coordinator. "But we have never had a great deal of success in trying to analyze these transactions, and get some useful information that top management can use to make decisions. We have these profitability reports that have to go to New York every month, you know, reporting by product line, by region, by manager, that sort of thing. Right now, that's a standalone system, it runs on ORACLE, it's not part of GPS. But we need

GPS data to make these reports meaningful, and the two systems just aren't linked together in any meaningful way. I suppose it goes back to the design question ... but it's often very difficult to prepare these reports the way we want them. GPS processes transactions OK, but we really need to link our informational needs a lot more securely to GPS."

Where a CIS Viewpoint May Hurt GPS

The most obvious flaws with a CIS conceptual approach to GPS are primarily organizational rather than technical in nature. Although most users agree that GPS does have its problems, they are now more or less "used to" the system. GPS is a comfortable environment insofar as it has a clear air of familiarity. A "new" approach to GPS in the form of a redesign to conform to a CIS conceptual viewpoint may be difficult for users to accept. Local management still has to be involved. Communication lines have to be strong. The most important consideration seems to be involving everyone possible, and establish formal liaisons wherever possible to facilitate user interaction and communication.

"I think that if when GPS had first been installed in the branches, the central design team had simply come up to us and said 'Hey, we're putting in a new system, we'd like your input', so many problems could have been avoided," remarked one middle-level manager. "There seemed to be two user camps, the 'technical' people, such as the system operators and data entry people, and everyone else, which included

managers such as myself. We (the second group) have always felt ignored by the technical people, even those within our branch. Only last month did someone come up to me for the first time and say, 'Hey, we're offering a class in LOTUS, do you want to come?' I personally hardly ever use GPS, but it seems to me that kind of communication would be a big help anywhere."

Another possible drawback of a CIS conceptual viewpoint is that it may almost provide too much autonomy and flexibility. One of the surprisingly positive aspects of GPS is that usually alterations are very difficult, thus prompting managers to make changes only when they are really driven by a need. One of the fears about replacing GPS is that people may get too "technology-happy," perhaps at the expense of productivity or at greater operational cost.

"Too much of anything can hurt you, and technology is no exception," commented one technical coordinator. "For example, we used to have a lousy electronic mail system; cumbersome, difficult to use. The only people who bothered using it were the 'whiz kids' who got a real kick out of playing with it. Now we have a great system that anybody can use. It's so good, in fact, that people are using it for everything, personal correspondence, doodling, etc., and not all of it is necessary. In fact, we have our country heads routinely tying up system resources sending electronic mail messages to people one or two doors away, when all they need to do is stick their head out the door and shout."

"My point is, GPS is a lot of trouble to fix or to

change. Therefore, we only make changes to it when the business really demands it, when it's urgent. My fear about replacing it with a sleek new system that has everything but the kitchen sink is that everybody will want to make changes for little bitty, insignificant things, because it's so easy and fun. We'll have people spending their entire day thinking up new colors for their screens, for example, at the expense of not getting the real work done."

Clearly, if a CIS conceptual implementation of GPS is not to cause harm, a great deal of time must be spent thinking about more than just the technical feasibility of the implementation. It is extremely critical to communicate with users at all levels of the organization, form liaison groups where needed, and provide clear policy and direction from the top as to what is expected out of the new system. Given the culture of the bank, this last point will probably be the most difficult to accomplish in practice.

"Top management policy doesn't get well communicated to lower levels around here," noted the middle manager. "It's not just true in systems, it happens everywhere. But the problem you get in systems is that mistakes are very costly, and that successes are often hard to quantify. Usually, the bank doesn't pat you on the back and say 'Hey, good job!' If they leave you alone, you figure you're doing well ... But everybody loves to point fingers and 'blame the computer' when something goes wrong, but when things go right, they seldom, if ever give the systems people the lion's share of

credit for their success. So I would say that we definitely would need more direction and support from top management next time around ... If I couldn't be reasonably sure of that happening, I'd rather stick with GPS."

Where A CIS Viewpoint May Create New Problems

There is no doubt that our CIS conceptual viewpoint has the potential to create new problems already discussed, such as a "technology-happy" attitude among users or further rifts between various management groups. Many of the organizational problems likely to be generated have already been dealt with here, such as alienation among middle and lower-level management, lack of communication, and "cultural divisiveness" between the technical and managerial user camps. However, a distinct new problem that some managers see is that more "advanced" systems such as a CIS system further drive a wedge between the "traditional" and "technocratic" bankers.

"I have been with this bank for a long time," stated one high-ranking vice president, "and I well understand that the banking industry is becoming more technologically driven, that we need to have sophisticated systems to respond to the changes taking place in the markets. But these innovations don't come cheap. They cost us, both in capital outlays and people resources. In the past, this bank used to be much more credit-driven than it is now. Our loan quality has suffered along with everybody's, but I think in our case we haven't done as well as we should in backing ourselves up

with a strong cash position, high liquidity, and return to investors. We are a successful bank, but not as successful as we could be in these traditional measures."

"I'm not saying that this is solely because we are spending too much money on technology, I think there are many reasons that contribute. But we do have to take a look at whether we are getting enough bang for our buck on these new systems, and more importantly, whether we really need them. Many people tell me it's sometimes hard to put a tangible value on new technology, and I know that there are many 'intangible' gains that result from advanced computer systems, but the costs of this technology are very tangible and very real. I personally think more attention has to be paid to measuring these variables, to putting the brakes on until we really understand the bottom-line impact ... but I believe I'm increasingly in the minority around here. There are certainly a lot of people in this bank who feel somewhat alienated by the 'technocratic' camp, and I hope that fancier and fancier systems don't continue to widen the gulf between people who understand and appreciate the technology and people who don't, especially if it's at the price of failing to continue our success in the long run."

On the technical side, many bank personnel who have a great deal of experience with GPS are concerned that the difficulty in obtaining competent technical expertise in some countries may become even more difficult with a more advanced CIS system. "I don't care what kind of system this

bank ends up getting," stated the operations manager, "you still have to have some local systems support team to keep up with and develop local applications. Fine. Now with GPS we have COBOL and CICS, which are difficult things to deal with, but we found the expertise. We also knew that our universe of needed knowledge would be pretty much limited to COBOL and CICS, or at least 'COBOL-like' environments."

"But when you talk about a system such as they're working on now, with fourth-generation machines, with relational databases, with ORACLE, with INGRES, you're talking really state of the art stuff. I'm not saying that's bad, on the contrary, it's probably long overdue from a systems standpoint. My worry is: where do I get the people to support and maintain these systems once I get them up and running here (in Latin America)? To compound the problem, we have a really "flexible" system in mind, which is great, except that it expands the range of systems that we potentially need to know how to support from a couple to maybe hundreds. It's a tremendous challenge to keep up with the technology even if you can devote every waking hour to it, and we can't ... I think that any new system you put in to replace GPS has to take this into account."

Where A CIS Viewpoint Can Increase Strategic Advantage

Up to this point, we have mainly been discussing the CIS conceptual view from a "better or worse than GPS" perspective, generally always in comparison with or in the context of the GPS environment. What we need to look at now

is what a CIS conceptual viewpoint can do on a global basis, even in facilities where GPS works well now, to increase the bank's overall global strategic advantage.

One issue that has been debated for a while by bank management is the question of consolidation of customers that deal with the bank on a more or less "globalized" basis, i.e. not just one branch, but several. The classic problem used to illustrate this is the "General Motors" problem. If General Motors deals with the bank in the U.S., and has a certain amount of money there, it may also have some money in France through its French subsidiary, some in Germany, some in Tokyo, some in Colombia, etc. Given that the account information for General Motors is spread out among many different banks, in many different countries, with many different computers, how does one go about answering the question: What is the total amount of money that General Motors has deposited with the bank, all over the world?

Virtually all bank managers interviewed agree that it is next to impossible to answer this question with today's systems, but also admit a solution to this problem is far from immediate. Many agree that this ability to consolidate is becoming much more critical, not just to find out account balances, but for many other applications as well.

"Our industry in general is becoming much more globalized," offered one division executive, "and for the bank to be successful in the future, it has to serve the

multinational clients as globally as possible. We don't have many strictly 'local' corporate clients anymore, most of our big bucks come from internationally strong companies with subsidiaries all over the world. There are lots of services we're trying to push at the local level, but we've now got just as many, if not more, global issues we're trying to address."

"For example, to take your General Motors example, say General Motors wants to borrow money from us, and we give them a global credit limit of \$1,000,000,000. This is the total they can have outstanding from us, all around the world. Now, suppose their French subsidiary has a local credit limit of \$1,000,000. OK. Now, let's say the head of GM in France comes to us and says, 'Hey, I know we only have a limit of \$1,000,000 locally, but we desperately need \$2,000,000 for a new plant we want to build.' We'd like to give them \$2,000,000 in France, it would provide good profit for us, but we don't want to take risks beyond what we feel is reasonable for GM."

"You still with me? OK, now if we had some way of knowing where the slack in the credit limits are, if we could find out where we could pull money from elsewhere in our bank, say Germany, to lend to GM, we'd be in good shape. For example, maybe out of that \$1,000,000,000 global credit limit, of which we've assigned \$1,000,000 to France, we've assigned \$2,000,000 to Germany, and they only need to borrow half of that, or \$1,000,000. In this case, it could potentially be a great strategic advantage for us to be able

to give \$1,000,000 of Germany's credit limit to France, while still keeping our global credit limit with GM where we want it. France would get the money they need, Germany would get what they need, we'd make more money, and everybody wins."

"The billion-dollar question is: How do we know where these possibilities exist? Right now, our systems can't communicate with each other across countries in many cases, so how does the GPS installation in France, for example, know what's going on in Germany? And even more importantly, how can we use the computer to manage our risk in this way, to make sure that our global credit limit isn't being violated? There are many other examples of this which I won't go into now, but I tell you, the strategic gains for the bank could be virtually endless if we could find some way of answering these questions."

The Issue of Global Risk

The above issue of global credit is really a subset of the more important overall issue of global risk management. Whereas global consolidation is essentially concerned with integrating transactions across country borders, risk management is much more concerned with the global integration of information. From a global perspective, it certainly can produce greater strategic advantage for the bank, both through global credit management such as we have just seen and global investment management. However, given

the autonomous nature of the bank's culture, in practice the value of global risk management may not be appreciated by or of concern to local managers.

"I think global risk management is definitely a double-edged sword in that it's both a very important issue and an insignificant issue," noted an investment group head. "On the one hand, it does offer many strategic advantages by enabling us to get the best possible utilization of our global credit lines, while reducing loan write-offs caused by inability to manage these funds correctly. It also has great strategic value in global securities, where we are trading in a variety of markets for clients located in many different parts of the world. Global risk management relies on the coordination of information across country boundaries to be successful, and I believe that any system that helps this will certainly help the bank as a whole."

"But the other side of the coin is - and I'm not pointing fingers at anyone, we're all guilty of this - that most managers at the local level either don't have the time to or really don't consider it important to manage global risk. What they really care about is their own local risk. Given that this bank is so entrepreneurially independent, and that people are encouraged to really produce, in many cases the bank manager in Germany really doesn't care about what goes on in France - let me rephrase that - doesn't find it in his or her best interests to worry about what goes on in France. And so you have a classic dichotomy - people may have a system that gives them great information about every branch

of our bank in a really coherent manner, with bells, whistles, and the like, but they won't use the information because it's not what they consider 'useful' for running their local businesses. Thus in this context, as you can see, global risk management may not add any real global strategic value to our business after all."

"Therefore, for global risk management systems to really add value to this bank as a whole, top management has to get involved, because they may be the only ones in the bank that really care about global risk. I think that top management also has to come to a decision: Is it preferable to have each branch manage its own individual risk, or is there some real gain to having global coordination? I think the latter is true in theory, but it won't work in practice unless the local managers can be made to care about global risk, for one reason or another."

Although it appears as though global risk may be a fairly insignificant concern to local managers compared to local risk, global risk may still be useful as an informational tool in that it may offer managers the opportunity to assess their own local risk management needs based on 'signals' that a global risk management system could potentially provide.

"I think when you say that we don't care at all about global issues, that's too simplistic," offered a local vice-president. "It's certainly true that we manage our business at the local level, and local needs often have priority. But

we're still a part of the bank. We still have to send reports to New York, we still have shareholders to answer to that care about the global picture. So we have to respond to global issues. But it's more than that. There are very real informational gains possible by knowing what goes on in other countries, particularly in our general region. Often, the information we receive about what's happening in other branches - and it's not just risk, there are other things too - can help us a lot in managing our own local risk."

"For example - I hate to go back to General Motors again, but bear with me - Let's assume, for the moment, that we're Germany. And say we're faced with what you just described, GM has a credit limit of \$2,000,000 with us, but they've only used \$1,000,000. Suppose, therefore, that we have extra funds to 'give out' in credit, but don't really care about helping France, we just want to manage our own loan portfolio as best we can. Even so, we know a lot from looking at what's happening over there. Why are they building a new plant? Does this reflect some overall global expansion? If so, will it spread to our branch? Will France's ability to extend credit to GM impact on our own ability to do the same? If GM-France defaults, what does that say in Germany? Will they default on us too? And so on, you get the idea, but the point I want to make is that we do care about information that is global in nature. We may not be managing globality per se, but there's no question that the existence of globality and global access to information makes our lives a helluva lot easier."

Global Impacts of a CIS Viewpoint

Clearly, issues such as global consolidation and global risk management represent only a few of the ways in which the bank could potentially garner global strategic advantage. The question to ask now is: Can a CIS conceptual viewpoint help the situation at all, and if so, how?

"I think anything that links the data resources of the bank together is bound to improve things," offered the outside consultant. "The global consolidation and risk issues that we're looking at here really aren't application-driven, they're data-driven. The problem the bank has is that nobody can figure out how to share the data in a way that makes sense. For example, when you look at all these subsidiaries in different countries, what should link them together? One obvious choice is the account number. Now the way the bank has traditionally assigned account numbers defies consolidation. This is so primarily because the account numbers to some degree depend on the account holder. For instance, names beginning with 'A' started with account number '1', names with 'B' account number '2', and so forth. I'm not saying this is exactly what they did, but you get the idea."

"Now, the rationale here was to make it easier to calculate an account number for a customer. Hey, why not, it's a mnemonic, it makes sense. But what happens if a company changes its name? Say that People Express gets

bought by Texas Air, so its accounts now fall under 'T'. Bang! There goes your neat numbering! The point is that account numbers should be arbitrary, they shouldn't have any connection to the client. Then you have codes that can be consolidated, they're amenable to change."

"The key to success in increasing global advantage through information systems is what we've said, a good design, independent applications surrounded by an integrating framework, with clean user inputs and data outputs. But that's not the whole story. We need a good DBMS, we need data management concepts, a set of procedures outlining how we're going to store and share the data. We need better engineering and productivity environments, we need to improve the way we go about designing systems within the bank, so that knowledge and concepts can be shared and learned from by all."

It appears that from a technical standpoint, the CIS conceptual model can address the above issues. As we have previously seen, the CIS conceptual model is a data-driven model, unlike GPS which is largely application-driven. It features clear control methods on the input and output ends of the system, shared data resources, and consistent external interfaces. And it provides linkages between data in various countries to enable consolidation and "globalized" data structures, while at the same time retaining independence of applications, enabling local management to continue exercising a high degree of autonomy and control over the functions of their systems.

While from a technical perspective, we may have a pretty neat package, organizationally things wouldn't be so neat and clean. Some of the reasons for this we have already observed: resistance to change, a perceived lowering of people sensitivities, poor communication, and the tendency of local managers to concentrate on their own business needs rather than global issues. But one manager pointed out what perhaps may even be a bigger stumbling block at the bank: the "me first" mentality. This manager observed that, because of the culture and personality of the bank, people are often reluctant to give up what they feel is rightfully theirs.

"It's essentially a cultural question," noted the manager, "but then most organizational problems are. One of the - I hesitate to say 'problems', call it 'values' - that we've always had here is the 'perform perform' atmosphere, where people literally have their futures hang on the number of visible contributions they make to the bank's success. Similarly, though mistakes are tolerated here, they're generally only tolerated if you can balance them with productive efforts. People know that they have to be seen producing, innovating, and contributing to advance."

"Given that's the case, it's not such a surprise to me that the GPS people in Europe were so reluctant to let other managers in on the development process ... Everybody wants credit for what's rightfully theirs, and usually doesn't want to share it with anyone. I'm not implying that we're

all egotists here ... but I do think that in an organization as large as this bank is, you have to scream fairly loud and often before anyone will pay attention to you. I believe that the biggest obstacle you will have towards installing a (CIS) system of the type you describe is that people don't always want to share their locally-developed applications and systems and databases with the rest of the organization, especially when they view what they've done as important feathers in their respective caps ... I'm not saying they won't share if pressed hard from the top, but deep down they don't like it. I don't like it. It's tough to share ideas with somebody else when you're not sure if you'll get credit for it ... Until that attitude shows signs of changing around here, I mean at the top, I don't think you will have a great deal of success with any global systems cooperation effort such as you're describing."

Summary - What Now?

We have seen that the CIS conceptual model has a chance for both success and failure at the Bank. Although technically it offers many advantages over GPS, it can also create a host of new organizational problems and concerns that may be difficult for the bank to overcome. So is replacing GPS a good idea? Maybe, maybe not. We are not intending here to say that yes, GPS should be replaced, or no, GPS should not be replaced. If we were to replace it with a CIS system, would that work? What do the bank's managers see as the most promising course for the future?

What are some of the solutions to the problems we have described thus far? We will use the next chapter to discuss these questions in more detail.

CHAPTER VIII

The Future: After GPS, What?

At this point in the research, we have uncovered and analyzed many of the problems and issues inherent in a possible transition from GPS to a CIS environment. To conclude our work, we would like to examine some possible "next steps" for the bank to consider. What kind of solutions exist to the problems thus far presented? What do managers at the bank feel should be done? And if we were to replace the existing GPS system, would a CIS environment be the answer, and what exactly should such an environment be like?

Clearly, when we are looking at solutions to the problems we have seen at the bank, we must consider that any technical and/or organizational solutions we propose may be limited in feasibility and/or effectiveness by the particular case we are dealing with. In other words, there may be some solutions, such as central planning, which may be extremely effective in certain kinds of organizations, but will clearly not work in the bank. We must also be aware that technical and organizational solutions are linked together. Any technical solution we consider has organizational implications, and vice versa. Thus, our analysis of these solutions has to consider not only the appropriateness of a specific solution to the bank as a whole, but also how its implementation within the

organization will impact both technical and organizational strategic advantage.

Some Possible Solutions

One factor that has continually cropped up in our discussions of the shortcomings of GPS is the lack of clear policies for changes and updates to the system as GPS has evolved. A good solution to this problem might perhaps be to form a committee responsible for communicating some of the local needs, with the goal of guiding the evolutionary process of GPS as the business changes and grows. In fact, the makings of such a committee already exist at the bank.

"We do have a GPS evolution committee," noted a technology supervisor, "but it's not without its problems. The biggest one is determining just who should be on the committee. Because of the way we operate, many of the local branch managers, whose input would be most valuable, can't take time away from the day-to-day operations to sit on the committee and contribute their ideas. On the other hand, we have generated a lot of interest from the systems people who are actually maintaining GPS, and we have several users of GPS who also would like to make their feelings known."

"On the whole, though we are a long way from actually beginning to sit down and make productive suggestions towards the evolution of GPS, I think it's an encouraging first step. We are trying to actively communicate with the user community as a result of this committee, and there's no doubt that this approach is a tremendous improvement over

what we've done in the past."

Improvement of communication between local branches as well as with Corporate seems to be an important part of making information systems run more smoothly at the bank. Most managers interviewed agree that a much more "iterative" process of communication would be helpful. Instead of ideas being generated by the users and then passed along to the systems groups without any further communication, it would perhaps be better to have more feedback between users and developers. A process which encourages users and systems development personnel to communicate back and forth several times, each time with slightly revised and better ideas, would be much more effective in fostering the types of systems that are really needed at the bank: systems which really fit user needs as much as possible.

"A problem that I see is that users aren't allowed to contribute as much to the systems development process as I think we should be," pointed out one account executive. "For example, we often ask to see 'sample' GPS screens before new enhancements are actually developed, so that we can comment on what features we like, troubles that we see, etc. We consider such feedback to be important to help us do our jobs, after all, we're the ones that actually operate the system, so we should have some impact on its design."

"But more often than not, we don't get the cooperation we expect, even from our own team. Now I know a lot of the reasons for this are that users in the past traditionally

haven't been willing to give their ideas, and the systems people just do what they think is best. Even now, there are some users who are reticent to comment on the systems, just because they feel nobody will listen ... I think a shakeup in the lines of communication around here is long overdue. Too many people on both sides are reluctant to speak out ... I am convinced we are missing out on valuable opportunities here because we don't communicate well. It's time for us to band together and get to know each other, really do something as a group for a change."

Although a CIS conceptual model, or indeed any other model of information systems, cannot directly force users to communicate better and/or more often, it may indirectly do so in that it greatly facilitates the process of communication within the bank, thus making it more palatable to people who have shunned heavy communication in the past because of its unreliable and time-consuming nature. By providing key administrative functions such as electronic mail and telecommunications networks, and by simplifying the processes of routing messages and data, a CIS environment can instill more trust in the computing resources of the bank, and convince more users and development personnel to "get together" and say what's on their minds.

"When you look at communication within this bank - I'm talking now about the physical process of communication, not just the willingness to communicate - it has historically been difficult and costly, especially across country boundaries," noted a regional line manager. "Phones,

telexes, and the like cost money, and we literally send thousands of messages each day, thus making the process even more complicated. To top it off, many regions in which we operate have very poor communications facilities, adding yet another difficult hurdle. Literally, it sometimes takes hours to get through by traditional means. To make things easier, we now have an electronic mail system and a lot of other bells and whistles that enable us to send messages fairly cheaply. It's definitely a step forward."

"But there's still a price to pay. It can still be very time-consuming, especially when our users aren't properly trained or sufficiently careful. The network isn't very 'idiot-proof'... Sometimes people don't use the system correctly, either through lack of training or simple error, and the system doesn't always compensate well. Messages can get lost, garbled, or destroyed so that we have to do them over, or the system just halts and won't let the users on. We need better control over this kind of stuff. We also need better controls on data security and what types of messages can get sent. We need ways of monitoring message flow through the system, and a network that gives us more capability to control this flow as required. We need a system that gives us flexibility in applications yet provides rigid communications and data control standards. In short, 'We need' a lot of things, but I kid you not - without them, I think we're in for trouble."

"Don't get me wrong, we now have one of the best

communications networks in the world. Most other companies haven't done better. But I think we can do better. I'd like to see more people using the system in the way it's really meant to be used. Not only do we need to spend more time and money training people, I also think we need to instill more trust in people that the system really can be made to work in this way. We need to alleviate the paranoia of the past, the ingrained reluctance to use the computer for critical communications. I believe the best way to do that is to make the system as technically perfect as it can be, so that you'd have to be a complete fool not to take full advantage of it."

A CIS conceptual approach, then, is a step farther along the road to providing a computing environment that provides for communication, for evolution, that instills trust, but still retains the essential elements of autonomy and independence that are so highly valued at the bank. Could such a system ever be "technically perfect"? Perhaps not, but it would certainly be superior to the 15-year old technology of GPS. Fourth-generation languages, expert systems, database management systems, and productivity tools are some of the "new breed" of technological innovations that, when applied in a CIS environment, may create very useful synergies within the bank.

"A lot of these innovations are widely perceived as being 'high-tech' and not suitable to a commercial banking environment," noted a technical advisory committee member. "But I would point out that you should look at the buyers of

these products. It's not just biotech companies or robotics firms. We see GM and Ford using executive support systems, we see General Foods installing expert systems for product planning, and nearly everybody is into database management. Why? Because they really do work. Companies in all kinds of industries, not just banking, are recognizing that these innovations can improve the bottom line. Executives spend more time thinking and deciding, and less time guessing and recalculating. These tools help bring key information directly to the fingertips of top management. They take a lot of drudgery out of the decision-making process. It's technology that wasn't available 15 years ago, but can and should be used now. You say 'high-tech,' I say 'good business sense.'"

Increased Control Resulting From CIS

Some other important benefits of a CIS approach to systems development at the bank revolve around the increased accuracy that results from the data control and message control functions of the model. By validating data up front before it even is "piped" to the applications, it is much easier to avoid compromising the integrity of the shared data resources. In addition, important messages between users and between different parts of the system are more easily communicated. Data and messages handled in this way can be more effectively monitored, and the "current" status of the database as well as the "age" of messages are more

easily known, a consideration that many bank managers think is extremely critical.

"One thing that would be extremely helpful is always knowing exactly where all our transactions are coming from and when they were originated," noted an internal controller. "We have lots of money moving through our GPS system, in a variety of currencies, every day. If one transaction is incorrect, we lose potential interest or investment gains on the money that we can't locate, as long as we can't locate it. So up-front validation is very critical. But in addition to that, given that we do have a bad transaction, it's really nice to know when and from where it originated so we can trace it properly."

"It's important to have a system that assigns the correct date, time, and origin to each transaction. Though GPS does this correctly between 70% and 80% of the time, in a bank our size, the 20% that we can't do right represents a lot of bucks we could potentially have trouble finding. So I really think we have to strive to reach that 100% mark as soon as possible ... It may be difficult from a technological standpoint to do this in all cases, I don't know for certain. But if we don't have the kinds of control we need over transaction errors, the system will soon stop. Dead. Because errors multiply very quickly when you're dealing with our volumes, and the process can't tolerate that for too long without breaking down."

It is apparent that the CIS conceptual approach to systems development at the bank can do much to reduce errors

and the costs associated with them, because it explicitly provides for greater control over messages and data, and therefore minimizes the likelihood that things will get too far out of hand. Such techniques as date/time stamping, message "tagging" at point of origin and point of arrival, and "intelligent" data entry screens that perform pre-validation functions are some of the ways that a CIS system might increase this overall control.

Another way in which a CIS model might be of help in this regard is through the support of data control functions such as concurrency control. These kinds of features not only improve data validation by preventing transactions from being "changed" from multiple locations simultaneously, they help ensure that the most "current" versions of transactions are always available to users of the system.

What of the Future?

Not only is there a wide difference of opinion among bank management as to whether GPS should be replaced or not, managers who do suggest replacement also have varying ideas as to what type of system should replace GPS. Most agree, however, that the exact types of hardware, software, etc. to implement are less critical as are the basic capabilities of such a system, i.e. what the system should really be able to do that GPS cannot.

Though not all managers agree exactly on these specific capabilities of any new system, there are a few common

threads. One is the issue of consolidation. It is thought to be very important in the future to be able to bring many bits of information from far-flung regions of the globe together in some centralized package. "It won't be just the 'General Motors' problem that we'll solve here," noted a regional vice-president, "we'll also be able to find out the answers to the questions of: What's really going on around here? What's the bank doing? How can we improve global opportunity? Those are the questions we'll need to answer in the future, to enable us to manage the bank as a global business, not solely as a series of regional businesses."

Another common thread is that key decision makers need to have exactly what information they need to make these decisions. Although consolidation will improve the quality of information, the issue is not so much concerned with consolidation itself as it is with what to consolidate. "We have to do a lot of soul-searching," noted a senior vice-president, "and ask ourselves: What is it that we really need to know? A lot of us don't have a concrete answer this minute, because we've never really thought about it. We run our business based on experience and instinct in many cases, it's hard to quantify every little piece ... Maybe we can't ever do so completely."

"But what this says on the systems side, I think, are some of the things you've talked about: good design, planning, strong communication and linkages with users. Next time we can't rush into things blindly. We can't assume we have a 'total solution' to everybody's problems, we have

to take responsibility for really finding out what's important to the business. One of the best lessons of GPS is that it showed us there's no substitute for being aware, for knowing what's going on. I think that will be a major difference in the systems we install here in the future."

The final common thread is that control of the systems development process from both a cost and efficiency perspective is viewed as critical. Learning how to size and price hardware, software, communications equipment, and other things will be important. Controlling the labor costs of development and maintenance are also considered essential. And the establishment of policies and means of communicating them through the organizational ranks is seen as a mandatory step before any future systems efforts can be successful at the bank.

"Some of us think the answer is downsizing so we can reduce hardware costs," observed a technology director. "Others think that 'regionalizing' to fewer data centers would be appropriate so we can reduce personnel costs. Both may be right, it depends on the country and situation. The policy of things, the methods of getting things done, that's the key as I see it. If we have a good policy, that at least is well understood if not fully agreed with, if we know exactly how we're going to go about deciding whether we want more data centers or fewer data centers or more mainframes or more PC's or less people or more people, or even if we want to keep everything the same ... as long as we control

how we do things and communicate the gospel, as it were, to the rest of the bank, we'll do OK. You ask me what I would do technically, I don't know, I don't think anybody knows, you don't know. It's really too early to tell. But organizationally, I would say let's talk policy. And let's keep talking policy, dammit, until we find one we can agree on, or at least accept as the way we're going to do things from now on."

We see that solutions for the future do exist. But most are anything but clear. The most important thing that is clear is that planning, design, and control are the elements most sorely lacking in the existing GPS environment. And much of the bank management concurs that good solutions will probably not be effective without greater attention to these elements in the future.

Will a CIS environment be the chosen direction for the "next generation" of GPS within the bank? Although we have seen that it can indeed improve global strategic advantage for the bank, its effectiveness will be directly dependent on the degree to which planning, design, and control are exercised in the development and implementation process. It seems as though the failures of GPS at the bank have not always stemmed from GPS itself, but often from a lack of attention to the above elements, from an inability to plan and exercise clear policy directions over the GPS development process. Despite all the problems that GPS has caused at the bank, it has made managers more aware of the dangers of inattention to planning, design, and control, not

just in the case of GPS, but perhaps in other cases as well.

With these concluding thoughts, it may be prudent at this point to summarize what we have learned, and perhaps suggest some tentative conclusions that may put the research as well as the case study into final perspective.

CHAPTER IX

Summary and Conclusions

So, after all these interviews and discussions, what have we learned? To put the answer to this question into the proper perspective, it is necessary to look back at our initial description of the composite information systems methodology (Figure 1). This methodology initially proposed that technical and organizational issues be discovered and clarified, and that we "map" the resulting problems onto both technical and organizational solutions. Thus we shall now attempt to summarize the main issues central to our discussions of GPS, both technical and organizational, and then outline the solutions that a CIS conceptual viewpoint may suggest as possible directions that the bank could take in the future.

Technical Issues and Solutions

The major technical issues we have thus far observed surrounding GPS vary widely by country and are perceived differently from manager to manager, as we have already seen. However, the major overriding points of agreement seem to be as follows: (1) lack of detailed technical documentation, making changes to GPS difficult; (2) 15-year old, largely outdated technology, plus a lack of technical expertise in some areas; (3) poor data validation and error-checking routines, making it difficult to adequately

compensate for human error; (4) lack of system planning and control mechanisms, increasing the difficulty of both satisfying local needs and maintaining a high level of trust in the data; and finally (5) an inherent lack of flexibility in the design and structure of GPS, making it difficult to reduce the costs of information and/or improve the quality of information provided to bank management. In other words, GPS defies attempts to increase the bank's overall strategic advantage.

As an outcome of our discussions of these issues, and the resulting insights into what a CIS conceptual viewpoint can provide, we can summarize briefly the ways in which these issues can be addressed. (1) Better documentation is clearly needed next time around, but the process of change can be simplified with a data-driven system. Having a system that relies on common databases simplifies the documentation process, because data descriptions and record definitions, usually the most critical documentation aspects, can be defined once and left that way. The only new documentation that need be written in this case is for the individual applications.

(2) New technology is needed: DBMS's, productivity environments, better communications networks, intelligent terminals, etc. The bank needs to get systems that "run themselves" better, that really manage information rather than merely process it. Then managers can spend less time on the technology and more time on the business.

(3) As regards the question of error checking and

validation, the CIS methodology provides message and data control functions to pre-validate data before it reaches the applications, as well as to provide concurrency and routing functions to make certain that the shared data resources are properly updated.

(4) Better planning is needed up front, top management has to involve the users more thoroughly. The establishment of design teams at the local level, as well as a user liaison mechanism to permit "iterative" change management is highly desirable. The systems development process has to be assessed with respect to both the effectiveness in meeting user needs and the feasibility of control over the systems that are put in place.

The most important issue to address is the last one (5), that any system that replaces GPS must help to increase strategic advantage. The most important characteristic of such a system is that it can provide for integration, independence, and evolution. As we have seen, the bank needs all three in abundance if it is to perform well in the future.

The best strategy, as has been noted many times thus far, is that the system must have a good design. Everybody's input must be recognized, and top management must be sufficiently involved to both communicate the significance of the system to local management and to provide the strategic directions necessary to assist in the evolutionary process. The system must be designed along these lines, with

local management providing the impetus for independent applications tailored to user needs, and top management providing the guidance towards a more integrated system, as well as the overall strategic evolutionary path that the system will have to follow in the future.

Organizational Issues and Solutions

Though what holds true with technical issues also applies to organizational ones: that there is much disagreement across a wide spectrum of managers and countries, there are again some pervasive cultural aspects that must drive any future information systems development and implementation process at the bank. They are: (1) a need for autonomy at the local level, a necessity for local managers to have as free a hand as possible in running their businesses; (2) a similar need for local flexibility, an imperative to respond quickly and accurately to local banking needs, without being hampered by systems that aren't flexible enough for local requirements; (3) a history of poor communication between top management and the rest of the organization, coupled with a lack of commitment at all levels to clear policies governing systems development and change management; (4) a general lack of trust in the quality of information, both at top levels and lower down; and finally (5) a "perform perform" atmosphere garnering a "me first" mentality, which leads managers to care very little about sharing their ideas or adding value to the bank as a whole, as long as their local operation is performing to their individual

expectations.

The issues raised above, as we have noted, have answers that are less clear-cut than on the technical side. Nevertheless, there are certainly some solutions we have discovered through our discussions that at least have a strong possibility of generating good results for the bank as a whole.

(1) The need for autonomy is addressed by fully independent applications. In this way, managers do not have to deal with the "core code" of GPS. Rather, an attempt is made from the beginning to find common data needs, while allowing managers to manipulate the data any way they wish.

(2) Flexibility is addressed in much the same way, with the addition that smaller systems that are better distributed seem to be the answer. Instead of having the strictly "modular" design of GPS, it may be more useful to have stand-alone, independent applications that are PC-based, but "plug in" to the common databases previously noted for any application required at the local level.

(3) Poor communication is very difficult to address directly at the bank, because correcting the problem requires both a stronger commitment by top management and local users. However, many people seem to feel that besides obtaining more technically advanced communications methods, management must become more people-sensitive, understand that the users really have a lot of valid ideas to contribute. In addition, the "technocrat" side of the bank

needs to be more aware of and responsive to the needs of those individuals that really aren't as "technology-driven" as the rest of the bank. An effective policy for systems development at the bank will also hinge on these points, for not only does top management have to improve communication to implement a policy that works, something that is so central to effective information systems planning at the bank will be less than fully effective if all users of the system are not allowed to contribute fully to the process.

(4) Trust in the system can be improved by proving that it works. It is clear that advances in technology can help instill this confidence, but there is more to it than that. Many bank managers feel that top management needs to play a more active role in convincing users to use the technology. Generally, if the higher-level people with a great understanding of the risks use the systems regularly, other people will feel more confident in doing the same.

(5) The hardest aspect of the bank's culture to deal with is probably the "me first" mentality. There are no clear solutions to this one that promise definitive results. In some aspects, the bank may really be better off doing nothing, for as we saw with the global risk issue, the bank may continue to be very successful if everybody maximizes their own business rather than thinking of the bank as a whole.

Probably the best of both worlds is to reward managers both for running their own businesses and for contributing to other parts of the bank. This approach opens up a wide

"spectrum" whereby some managers may elect to continue focusing on their own branch, while some may divide their time more equally between projects for other branches. In this way, people get credit and recognition for doing both, and the result is a better balanced bank environment, with people feeling no pressure to share ideas and resources, but not losing control over their own branch if they do. The key ingredient again is an attitude on the part of top management that supports this type of environment and makes it viable. Without this ingrained commitment at the top, this kind of change, if it is implemented, will not be successful in the long run.

We have thus completed a brief summary of the major "problems" and "solutions" that we see in the bank. Table 1 provides a tabular summary of these conclusions to assist in clarifying our final points. It is now perhaps appropriate to tie these conclusions together and offer some final points, to provide the bank with a more definitive path for the future.

The Bottom Line

In the final aftermath, is there anything that the bank can do to increase its strategic advantage through composite information systems? Clearly, yes. There is no doubt that a CIS conceptual model can improve things at the bank. The bank will be able to develop better control mechanisms, it will have increased globality of data, and the ability to

TECHNICAL

"PROBLEMS"	"SOLUTIONS"
1. LACK OF DOCUMENTATION	1. DATA-DRIVEN SYSTEM UNDERSTANDING NEEDS
2. 15-YEAR OLD TECHNOLOGY LOW TECH. EXPERTISE	2. NEW TECHNOLOGY "MANAGE" INFORMATION
3. POOR VALIDATION TOO MANY ERRORS	3. MESSAGE / DATA CONTROL
4. LACK OF PLANNING / CONTROL	4. LOCAL DESIGN TEAMS USER LIAISONS
5. LOW FLEXIBILITY / STRAT. ADV.	5. GOOD <u>DESIGN</u> / TOP-LEVEL INV.

ORGANIZATIONAL

"PROBLEMS"	"SOLUTIONS"
1. NEED FOR LOCAL AUTONOMY	1. FULLY INDEP. APPLICATIONS NO "CORE CODES"
2. NEED FOR LOCAL FLEXIBILITY	2. SMALLER SYSTEMS "PLUG IN" TO DATABASE
3. POOR COMMUNICATION	3. <u>TOP</u> MGMT. INVOLVEMENT
4. LOW <u>TRUST</u> IN INFORMATION	4. PROVE IT <u>WORKS</u> TOP-LEVEL ACTIVE USERS
5. "ME FIRST" MENTALITY	5. ? - DO NOTHING BETTER INCENTIVE SYSTEM

Table 1. Summary of Technical and Organizational Issues.

retain the local autonomy and independence to which it has become accustomed. But the above statement is too simplistic. In practice, there will be certain real costs associated with the transition to a radically new breed of system, a system that despite its powerful potential, will only be successful if it can be well integrated into the culture of the bank.

Any time an organization needs to change its way of doing things, either internally or externally, there are always upheavals, sometimes with drastic consequences. The organizations that will be most successful in dealing with these crises will be those that excel at change management. This is a more succinct way of saying that the bank needs to where it wants to go and what the purpose of the change is, and develop the appropriate and necessary guidelines for control, design, and planning for evolution of the change.

In other words, regardless of the approach that the bank takes to systems development in the future, and irrespective of whether it remains with GPS or uses a CIS viewpoint, or tries something completely different, the key to the bank's success will be its ability to manage and control change. If the bank can do this effectively, it will not only be able to use information systems to achieve greater strategic advantage, but to improve the potential for future strategic gains on a variety of dimensions.

What This Means

Change management is not an easy animal to define, and it

is difficult to state exactly how the bank should approach the management of change. Although there are many ways in which we might approach this issue, it seems that most of them would eventually take into account one key point: people at the bank are largely motivated by what is in it for them. In other words, when change must occur within the bank, it will be most successful and most enthusiastically received by the individuals or groups that benefit most from the change.

Given that this is so, it seems that a good change management strategy where information systems are involved would be to develop systems that potentially reward the most people possible. Of course, these efforts have to be assessed as well in terms of the costs associated with developing these systems in the first place. But the question to ask if some group within the bank desires change is to ask: Is there some way that this change to the system could benefit other groups as well?

For example, if we are trying to develop a new global risk system at the bank, we may have difficulty justifying it on the grounds that many managers don't really care about global risk, or that not enough care to make it worthwhile in terms of the costs involved. However, a global risk system, in addition to managing global risk, might also be able to manage global custody, or global credit, or global profitability. It is very possible that the same managers who think that global risk is unimportant may at the same

time feel that global profitability is extremely important. It doesn't really matter that the system also provides global risk, it's an extra "feature" that they just don't happen to need. The key is understanding where the potential lies to create these kinds of synergies on a global basis. This is something that many local managers may not understand, the linkages may not be clear to them between global risk and other types of functions, for example. In this case, the burden is on top management to manage these linkages, and provide the proper impetus to enable the bank to manage change effectively.

If the bank's approach to change management is such that it directly caters to the inherent autonomy and independence within, the need for people to satisfy their own local priorities first, the result will be global change brought about through coordinated local change. Change management brought about in this way will not only enable the bank to increase global opportunity and strategic advantage, but additionally maintain the essential cultural elements that have made the bank such a success to this day.

FINIS

CHAPTER X

Bibliography

Primary References:

1. Madnick, S. E., Wang, Y. R., "Strategic Applications of Very Large Databases Through Composite Information Systems Design", 1987.
2. Frank, William F., Madnick, S. E., Wang, Y. R., "Testing A Conceptual Model for Distributive Processing: One Organization's Experience" (draft), 1987.
3. Donovan, J. J., Madnick, S. E., "A Technological Model for Distributive Processing", November 29, 1982.
4. Keen, Peter G. W., Bronsema, Gloria S., Zuboff, Shoshanna, "Implementing Common Systems: One Organization's Experience", CISR Working Paper #80, August 1981.

Secondary References:

- Bullen, Christine V., Rockart, John F., "A Primer on Critical Success Factors", CISR Working Paper #69, June 1981.
- Frank, William F., "Two Large Historical Databases" (draft) 1987.

INTEGRATING SYSTEMS FOR FINANCIAL INSTITUTIONS SERVICES USING COMPOSITE INFORMATION SYSTEMS

MARIA de las NIEVES RINCON

When designing and developing composite information systems for connecting computers around the globe it is important to recognize that the principles of autonomy, integration, and evolution need to be mediated in order to derive maximum advantage.

Autonomy refers to the capability of treating each system as a discrete entity that can be developed, operated, and managed independently of the other systems. Further, each system must maintain a high level of processing integrity to complete its functional responsibility as if it were totally independent of all other systems. In a decentralized environment, where each organizational unit has its own set of priorities, cultures, and objectives, it is important that efforts at integrating computer systems do not compromise the ability of individuals to operate independently in their respective spheres of activity. Autonomy can be implemented in none, some, or all of six categories:

- (a) Hardware Control -- This refers to centralization or decentralization of processing capacity;
- (b) Operational Control -- This relates to control over operations of computer center;
- (c) Transaction Control -- This includes functions such as data entry, verification, and updating of transactions;
- (d) Software Development -- This encompasses all aspects of software development;
- (e) Data Control -- This deals with development, control, and maintenance of databases; and
- (f) Management Control -- This covers settings of standards, planning, and acquisition.

Integration refers to issues of coordination and consistency across systems. While integration provides several potential benefits, it also involves a high initial cost. For example, integration of several existing databases is usually a major exercise. However, by separating the data from the application, future changes to the data will not require a change in all applications.

Evolution refers to the ability of the systems to accommodate growth. The architecture should be flexible enough to allow capacity to be augmented. After existing systems have been integrated, the users frequently come up with a new and vastly expanded list of desired reports and summaries. This in turn implies a higher level of computational speeds.

The above issues are examined using a case study of a large multinational organization with significant operations in all parts of the world. The case study identified that operations of this organization are hampered by a lack of adequate foresight in many of the areas mentioned above.

Introduction

In this thesis, the case of a very large multinational bank is analyzed. The focus and scope of the analysis is on information systems technology, and in particular, on the implementation of Financial Institution systems. These systems support the products and services provided by the bank to its Financial Institution customers in US Dollars. These customers are banks themselves both, domestic and international. Corporations and individuals are served by other organizations within the bank. The bank has over 8,000 Financial Institution customers. The volume of transactions processed per day in real time by some of the FI systems can be as high as 85,000. These transactions involve billions of dollars. For instance, the total amount of money transferred through the bank's Funds Transfer System can be up to \$250 billion per day. Needless to say, the accuracy, reliability, risk control, and ability to provide a continuous service are of primary importance for the customers, and the bank.

The analysis and evaluation of the integration capabilities of Financial Institution Systems is done using the Composite Information Systems model [1]. The case of this bank is of particular interest since, first information systems technology plays an increasingly important role in the ability of a bank to deploy its products and services. Second, the

bank has a very decentralized organizational structure (see figure 1) which has had an important influence in the development and implementation of information systems. These systems have been developed independently following the decentralization trend, and the result has been in some cases, a collection of disparate systems. Third, the financial industry is undertaking continuous external changes, and the Financial Institution systems need to respond quickly to the changes imposed by external forces to keep its competitive advantage.

The issue of connectivity and integration among these disparate systems has also become increasingly important. Obsolete practices such as, tape hands-off to communicate among systems are still used in the bank. Tape hands-off require manual intervention, and slow down the communication across systems. Disparate information systems are usually accompanied by disperse and redundant databases. The dispersion of data has an impact on performance, operational costs, duplicate maintenance, inconsistency of data, and real time checks against the bank books. All of these are of critical importance for the transactions processed by the Financial Institution systems group. However, the integration of the systems must be achieved while still maintaining some of the flexibility that was obtained by the decentralization of systems development and operations.

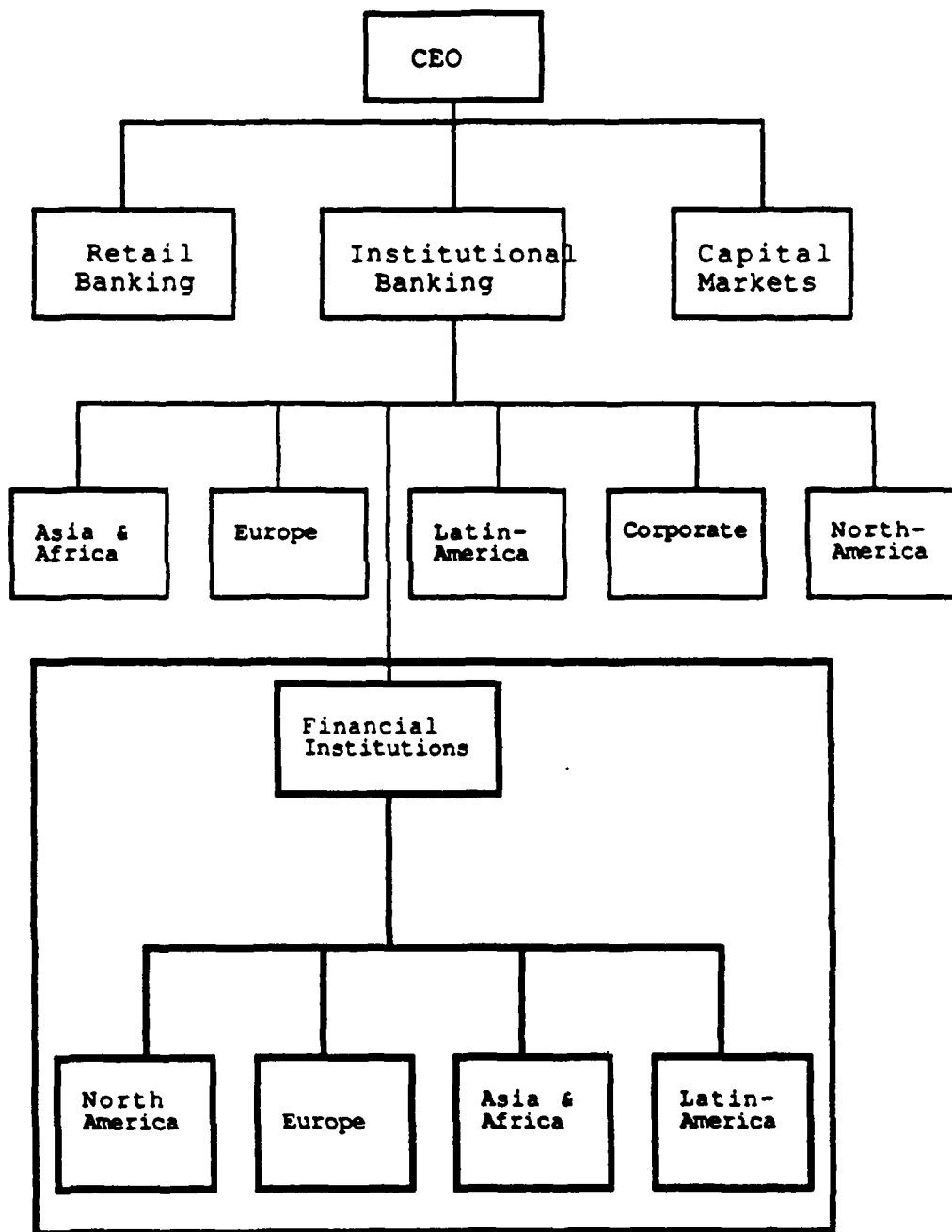


Figure 1

The bank's organizational structure

Therefore, three attributes have been identified as critical in the development and implementation of Financial Institution systems: integration, autonomy, and evolution. Integration, refers to the ability to connect disparate systems, which in most cases have disperse and redundant databases, called "shadow databases." Autonomy refers to the independence of systems, which provides flexibility in their development and operation. And evolution refers to the ability of the systems to evolve to accomodate either internal or external changes.

A conceptual model, which is described and analyzed in this thesis, was developed in 1982 as a framework for the implementation of Financial Institution systems. This model attempts to mediate the conflicts of trying to meet the three attributes mentioned above. This paper evaluates the current implementation of the Financial Institution Systems which have followed the conceptual model guidelines. The integration and connectivity capabilities of the Financial Institution systems are further evaluated identifying strengths and weaknesses from the technical perspective, and potential future problems that may arise from their lack of integration. Organizational issues are also considered since they are an important factor in the successful implementation of technology, particularly in the bank.

The thesis focuses mainly on technological issues, and the following questions are addressed throughout the thesis. 1)

What systems are in place to serve Financial Institutions?, 2) What technologies are being used?. 3) How do the systems conform to the goal of CIS?. 4) What are the constraints imposed by the current systems to achieve integration?.

After evaluating the current implementation of Financial Institution Systems, alternative solutions are considered in order to achieve the goal of integration.

Chapter 1

Composite Information Systems

1.1 The Composite Information Systems concept.

The Composite Information Systems (CIS) concept is presented as a methodology to map strategic applications to appropriate solutions using information technology and organizational theories [1]. A Composite Information System is defined as a system which integrates independent systems which reside within or across organizational boundaries. CIS thus provides a top down process by which the strategic goals are identified and are linked to technological innovation and organizational structure (see figure 2). The CIS model recognizes that there are both technical and organizational obstacles that constrain the ability to connect systems that have been developed independently. Some of the technical problems encountered are, heterogeneous hardware configurations that do not communicate with each other, different software environments, independent databases, and different database management systems. Some of the organizational problems that can be identified are, lack of standards, great degree of divisional or departmental autonomy, and lack of communication among groups. These obstacles jeopardize, to a greater or lesser degree, the feasibility to connect these systems. The purpose

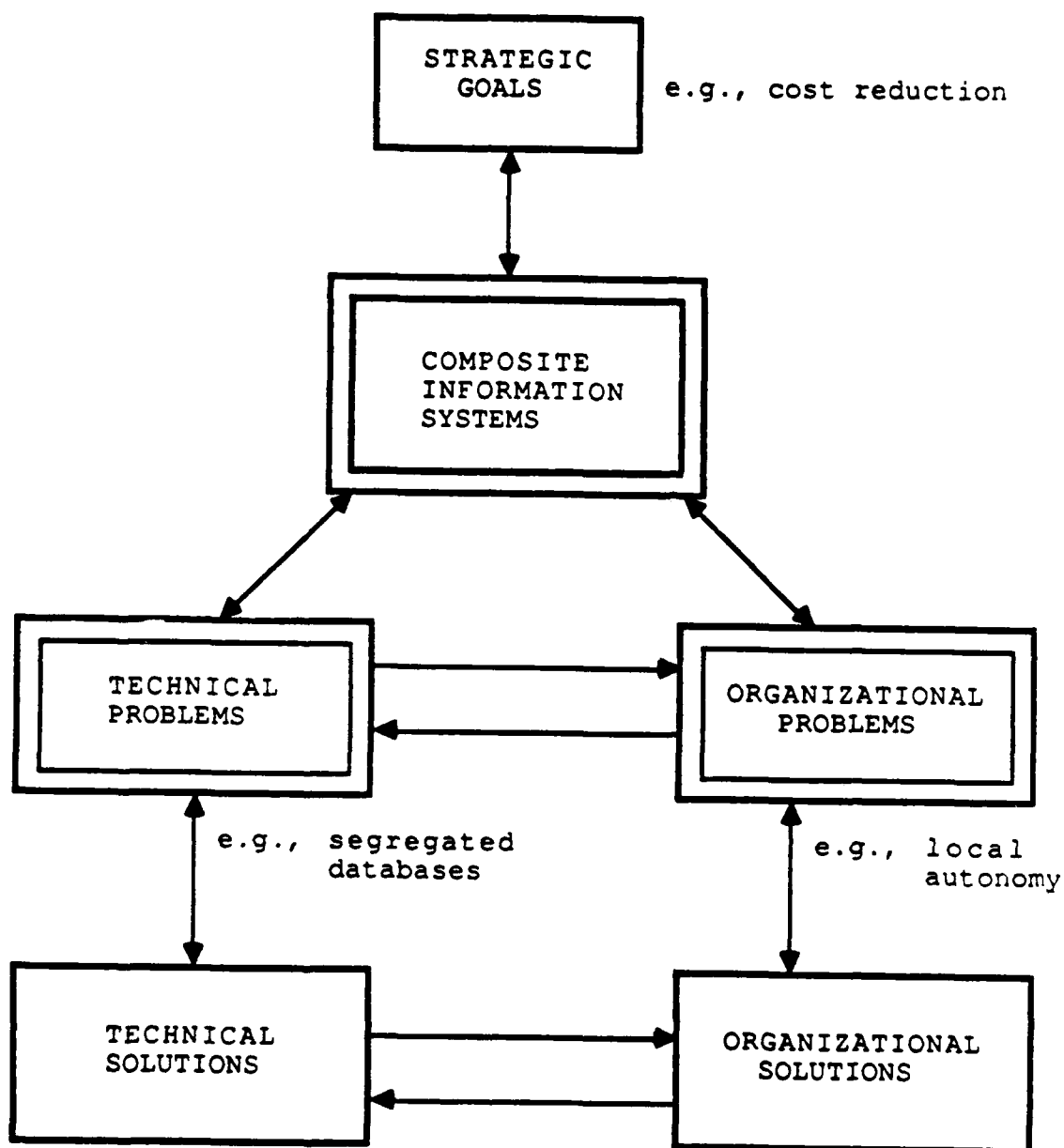


Figure 2

Composite Information System model

of the CIS concept is to help find technical and organizational solutions to achieve the interconnection of disparate systems. The interconnection of these systems can reduce costs by, exploiting economies of scale through the sharing of resources, eliminating human intervention and tape hands-off, reducing the number of errors, and eliminating duplication of functions, thus reducing operational costs. On the other hand, interconnection can make available distinct or complementary products not offered by the competition.

1.2 POTENTIAL CIS APPLICATIONS.

Madnick and Wang [2] have identified four categories of potential CIS applications: 1) Inter-organizational applications. These applications involve systems that go across organizational boundaries, such as linking customers and suppliers. Examples such as, American Hospital Supply system [3] and the Airlines Reservation Systems [4] are found in the literature and belong to this category. 2) Inter-functional applications. These applications involve systems that interconnect applications from different functional areas of the organization. For example, both the purchasing/inventory control system and the accounting system of a corporation need information regarding the materials that are delivered to the warehouse. The former needs the information to keep inventory control, the latter for invoicing purposes. However, it is not unusual to gather and

enter this information independently in each of the two systems. In this case, the failure to link these two systems leads to the duplication of the collection of information and data entry functions, and to potential discrepancies on inventory information. 3) Inter-product applications. The interconnection of applications that support independent products have become increasingly important to provide a single product that consolidates existing products or services. The Merrill Lynch's Cash Management Account example found in the literature [3] belongs to this category. 4) Inter-model applications. These applications involve the linkage of different models, policy models, econometrics models, etc., which may have been developed under very different environments.

All these potential applications require integration among intra-organizational or inter-organizational systems. The integration will ultimately accomplish cost reduction or product differentiation, as it was illustrated in the examples given above.

The advances in data base technology and data communications help to accomplish the goal of systems integration. However, integration may be jeopardized by historical and/or organizational reasons. Some systems were designed years ago, as independent systems and with old technology; thus, many of these systems have combined the data and the application into

a single package. On the other hand, some organizations have broken down the business of the firm into autonomous units that can be managed as viable and isolated business and where the goals of the management are confined to the unit. Thus, in many cases, they have distributed the processing power. As a result, independent systems, computing centers, and databases exist. Another problem is that the management of the different units may not be willing to relinquish the power and control they have gained through decentralization. CIS recognizes that these obstacles exist and that the goal of integration for these systems involves both technical and organizational solutions to solve the obstacles that interfere with integration.

Chapter 2

CIS principles: autonomy, integration, and evolution

The case of the Financial Institution systems being analyzed is of special interest since, first, most of the systems have been developed independently. Second, the organizational structure of the bank is highly decentralized, and the flexibility provided by local autonomy is very important for the management of the bank. Finally, the products and services offered to Financial Institutions vary with time and the systems need to conform to these changes.

Thus, CIS recognizes that the principles of autonomy, integration, and evolution need to be mediated to keep the advantages of each one of them. A discussion of these principles in the context of the Financial Institution systems is presented in this chapter.

2.1 AUTONOMY.

Autonomy refers to the capability of treating each system as a discrete entity that can be developed, operated, and managed independently of the other systems and where each system must maintain a level of processing integrity to complete its functional responsibility as if it were totally independent of other systems. Autonomy is a very important attribute that

matches the organizational structure of the bank which is highly decentralized, and where control, responsibility and accountability have been distributed to the lowest management level of the organization. From the systems perspective, autonomy refers to the ability of individual managers to control the resources they need to support their business needs.

The autonomy discussion can be related to the centralization vs. decentralization dilemma. Different authors have discussed this issue. In particular, Rockart [5] presents a framework for the analysis of centralization vs decentralization. The framework identifies three dimensions where decentralization can be implemented namely, systems operation, systems development, and systems management (Figure 3). Although this framework is very useful, it is too general in the context of the bank. Therefore, based on Rockart's framework and taking into account the characteristics of the bank, autonomy has been identified in six categories or dimensions which are illustrated in figure 4. Autonomy can be implemented in none, some, or all of these categories. The six categories are:

1. Hardware control. This category refers to the centralization or decentralization of processing capacity. The degree of centralization can be so much as having a large central hub that serves the whole organization to the

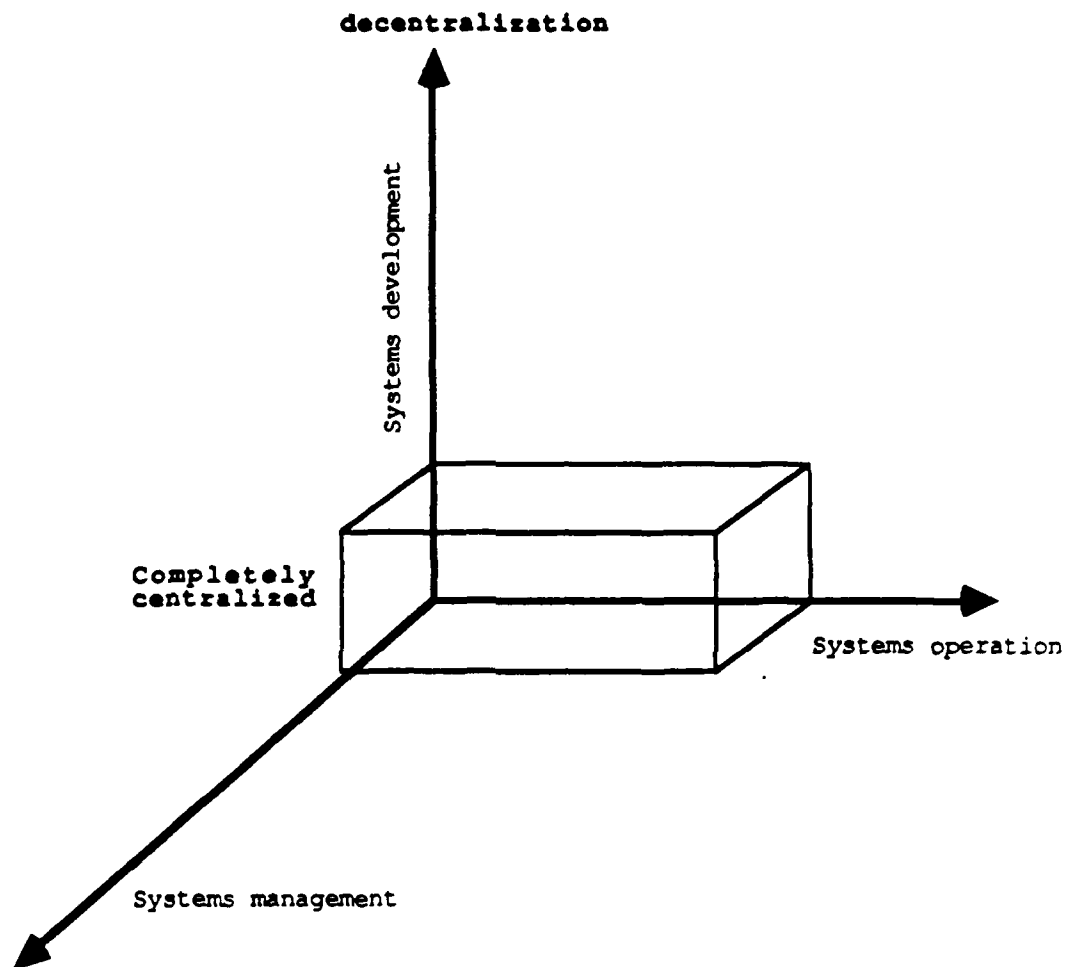


Figure 3

Dimensions of Centralization/Decentralization [5]

The Financial Institution systems as of 1986

other extreme where each single unit has its own processing power capacity.

2. Operational control refers to controlling and running the operations of the data center. This category covers a range of functions, from hardware and operating systems maintenance to capacity planning. Although operational control is usually tied to hardware control (first dimension,) this need not be the case. For example, even if hardware control (i.e., capacity power) has been distributed to sub-organizations, the operations can still be run and controlled by centralized systems staff.
3. Transaction control refers to the management of the unit's transactions. This category includes functions such as, data entry, verification, and repair of transactions.
4. Software development refers to the design, implementation, and maintenance of software for applications. Centralized software development is done by central personnel staff. On the other hand, software development can be moved to the units to be closer to the users where it will be done either in-house or by contracting outside firms.
5. Data control refers to the development, control, and maintenance of the database(s). Centralized data control does not mean that all data should be kept in a single file

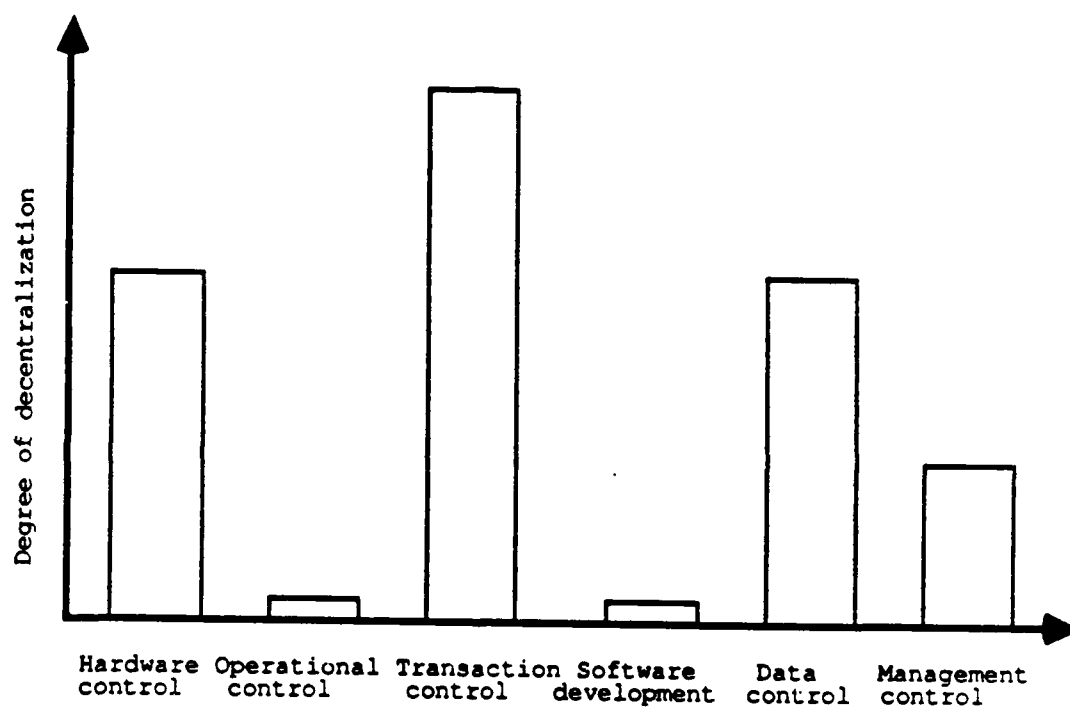


Figure 4

Centralization vs. Decentralization dimensions

The Financial Institution systems as of 1986

but that the the control of the data should be centralized, e.g. propagate updates, set standards to get access to data, etc. Many organizations are developing a data dictionary in which information about the data is stored and the data is standardized to facilitate communication among applications. Other organizations are moving towards distributed databases which allows the access to data contained at different locations.

6. Management control. Within this category, the functions performed are: the setting of standards, planning for data processing, hardware evaluation and acquisition, and decision making regarding the projects to be implemented.

A high degree of autonomy simplifies the task of managing each division. However, it presents some disadvantages: duplication of functions, inefficient use of resources, reduction of economies of scale, independent and redundant databases, and lack or poor communication among systems, such as tape hands-off.

As it will be discussed later, the systems in the bank have been developed independently having as a result independent databases. However, a lot of the information contained in the different databases is common or redundant across databases, creating "shadow databases". As a result of the shadow databases, updates need to be propagated to avoid

inconsistencies in the data contained in them. In some cases, consistency of the information is of critical importance, as is the case with a customer's account balances. This information is used by several applications, each of which has its independent database. So, procedures need to be set up to perform the updates propagation either in real-time or batch. If it does not need to be real time, a file with all the updates can be sent daily. In many cases, these procedures require human intervention, either "tape hands-off" or by entering manually the changes to the database. Otherwise, updates can be propagated automatically, exchanging messages across applications. If the updates need to be done in real time and there is a large number of those, this approach can result in an overload of messages. In all these cases, software needs to be in place to take care of the propagation of updates.

2.2 INTEGRATION.

Integration refers to the coordination and consistency across systems. Some general characteristics of integrated information systems are, to share common data, to enable communication among systems, and share resources. Integration provides several advantages over independent systems. It allows reduction in the duplication of functions, more efficient utilization of resources, more consistency across systems, and transparency in the use of different systems.

However, it also requires a higher degree of complexity to coordinate the development and operation of all these systems. Integration can vary in degree, one extreme would be to have a single processor where all the systems are hosted. In this case, other factors such as, response time, ease of migration to a larger system, and lack of flexibility to run some applications arise. The bank has an example of a such a system running in its overseas branches [6]. However, other organizations have followed the trend towards distributed processing given the better price/performance ratios for hardware, and the cost reductions and improvements in data communication and database technology. These organizations are finding a need for the integration of their systems. This last case holds true for the bank being evaluated.

The reasons for achieving system integration in the bank are several: to provide more consolidated services to the customer, to improve the internal operations of the bank, and to reduce operational costs. To keep its competitive advantage, the bank has an increased need to offer consolidated products that cut across different products and services. Systems integration is especially important to the bank since the systems are highly independent of each other as a consequence of its organizational structure.

Integration is also important for the internal operations of the bank, to better monitor transactions and to reduce risk

exposure. An example where globally integrated information is critical for the exposure of the bank is the monitoring of high risk economic situations. For example, it is necessary for the bank to know the exposure of Mexican pesos not only in their Mexican branch but its exposure for all branches worldwide. Integration is also important to check overdraft and balances against the actual bank books.

Standardization of some of the dimensions, hardware, software, programming languages, data, communications, and external interfaces is a way to help accomplishing the integration goal. Standardization of hardware, software, and data provides the basis for easier integration even if connection of current applications may not be a foreseeable need. Standard hardware and software will make it easier to interface applications when that need arrives, and standardization of data, (e.g. data definitions,) will ease the interchange of data among different applications.

There is a trade-off between the cost in developing vs. maintaining systems. Although integration usually requires a higher cost up-front, it will reduce future maintenance costs. For example, if a shared database is developed for all applications that need common data, higher costs and longer time will be spent for initial development (ie., complexity, coordination) but there will be future gains in terms of maintaining the databases (e.g. consistency of data).

Maintenance of disperse databases usually involves, tape hands-off, human intervention, and flow of messages across applications. Another example is provided by separating the data from the application by using data base management system technology. As applications will be separated from the data, future changes to the data will not require to change all the applications that have access to the database.

2.3 EVOLUTION.

The last CIS principle is evolution. This term means that the systems should be capable to accomodate change. The processing requirements within the bank are constantly changing by the introduction of new products, changes in the existing products, and changes in the external business environment, such as deregulation, improvements in processing methods, and changes in technology. Therefore, components within the system must be modular to adjust to change without disrupting other systems, and to minimize complexity in the evolutionary process.

Evolution also refers to the ability of the systems to accomodate growth. The processing capacity for the systems should conform to increases in the demand of the products offered to the Financial Institutions. Therefore, flexibility in the architecture for augmenting capacity should be taken into account. This implies flexibility in both dimensions,

hardware, and software.

Evolution is sometimes constrained by hardware and software limitations. For example, even if a DBMS would be desirable to ease the evolutionary process, it may not be feasible to implement due to performance issues. The complexity of systems in place are also a constraint on the ease of evolution. Adding functionality may take several months to be implemented. Other considerations such as, ease of migration should also be taken into account, e.g., if hardware runs out of capacity, the ability to migrate towards more powerful hardware (i.e., processor or peripheral devices) is a necessity. These issues need to be considered in the initial development to avoid future costly maintenance to the systems.

An example where evolution is important is the ability to provide the customer with integrated information. For example, a customer may want to have one consolidated statement with all his/her business transactions, such as loans, checking activity, funds transfers, etc. Although this product may not be demanded today, the systems in place should be able to evolve in order to provide this integrated service, whenever needed.

Chapter 3

The Financial Institution products and services

This chapter presents a description of the systems developed by the Financial Institution Systems Group (FISG) to support the products and services that are offered to the Financial Institutions customers [7]. These customers are defined as those which are themselves banks, including both, domestic and international banks. All the transactions handled by FISG are in US dollars. No multicurrency transactions are handled by this group. A summary of the FI products is presented in figure 5. As mentioned previously, the scope of this thesis will be limited to analyze and evaluate the systems developed by FISG.

The Financial Institution Systems Group is based in New York and gives support to the business divisions which offer products and services to Financial Institutions for all the transactions processed in US Dollars. The functions served by the systems group can be classified as either business banking functions, such as Funds Transfer, Letter of Credit, Cash Management, etc; or supporting management functions, such as accounting analysis, and reporting.

A brief explanation of the products and services that are offered to the Financial Institutions customers follows:

Product	Description	Average # transactions per day	# of customers
Funds Transfer	Move funds electronically in US Dollars to domestic or international beneficiaries	60,000	8,000+
Cash Management	Customers are connected electronically to the bank and can perform funds transfers, balance inquiry, and transaction history inquiry	10,000	900+
Demand Deposit Accounts	Automated accounting and reporting system for customers	50,000	8,000+
Trade Services	Letter of credit operations and collections	1000+	8,000+
Inquiry/Investigations	Find causes of discrepancies between customer's and bank's records for funds transfer operations	1,500	8,000+

Figure 5

Financial Institution's products

1. Funds transfer. The Funds Transfer service allows customers to electronically move funds through the bank to domestic or international beneficiaries. The bank has direct, on-line access to Fed wire, S.W.I.F.T., C.H.I.P.S., and Bankwire/Cashwire. It also accepts input from the Cash Management Account, structured or unstructured format telex, mail, phone or facsimile. The funds transfer operation of this bank is one of the largest in the world. Besides having 8000+ Financial Institution customers, the bank serves as an intermediary for other Financial Institutions and therefore needs to keep information of about 20,000 direct and indirect customers. It handles an average of 60,000 transfers operations per day. However, in peak days over 80,000 transfers operations can be generated, which represent about \$250 billion in funds transfers processed in one business day. The demand for funds transfer services has experienced a continuous growth of 25% per year.
2. Cash Management Account. The Cash Management Account is an online, interactive electronic reporting and payment system for US Dollar accounts in New York. Customers are electronically connected to the bank and can obtain balance information, access a 45-day transaction history, send funds transfers, track a letter of credit and initiate letter of credit reimbursement authorizations and ammendments. The average number of transactions for the Cash Management Account is 10,000, approximately 4,000 of which are funds

transfer operations, and 6,000 are inquiries. There are 900+ customers for this product, and this number is expected to grow to 1,700 customers by 1990.

3. Demand Deposit Accounts Statements. DDA statement is the bank's automated accounting and reporting system for customers holding demand deposit accounts, depending on the customer's needs. Account statements are generated monthly, bi-weekly, weekly, or daily. This statement can either be descriptive, showing the detail of each item; or non-descriptive, summarizing the transactions. An account history statement is also generated for customers. This statement shows all overdraft charges and refunds of interest due to overdrawn accounts and backvalue adjustments. The Demand Deposit Account system has 8000+ customers (accounts) and handles 50,000 postings per day.
4. Trade services. This includes letter of credit operations, and collection. Letters of credit issued by a bank on behalf of its client, give a designated beneficiary the right to draw funds in accordance with the stipulated terms and conditions in the Letter of Credit. Trade services handle 1000+ transactions per day.
5. Inquiry/Investigation support. In financial transactions, a large number of discrepancies occur between customer records and bank records. Customers inquiry about the

source of these discrepancies and the bank is responsible to find their causes using the bank's records, and to fix the discrepancy if the bank is in error. This kind of inquiry is called an investigation. At the moment all the investigations involve funds transfer transactions. The inquiry/investigation division performs an average of 1,500 investigations per day.

The systems that support these products have been developed independently and there is a one to one match between the products described above and the systems that support them. The technical, organizational, and historical reasons for these independent systems, and their technical implications and connectivity capabilities will be analyzed in the following chapters.

Chapter 4

A historic overview of Financial Institution Systems

The Financial Institution Systems (FIS) have evolved through different stages. From a very centralized systems structure in the early 1970's, they changed to an extremely decentralized one in the middle 1970's. In the 1980's, they are implementing a new architecture (explained in the next chapter), that contains elements of both centralization and decentralization. This chapter discusses these three stages in the evolution of the Financial Institution's systems.

In the early 1970's, the Financial Institution systems followed a very centralized structure. All systems were big, complex, and hosted in one IBM computer. The control, maintenance, software development, operations, and management of the systems were all centralized and handled by the Information Systems Group.

In the 1970s, the organizational structure of the bank was very decentralized, with autonomy and control given to the lowest management level in the organization. The objective of this structure was to allow for rapid change and encourage creativity. However, the systems in place, which were highly centralized, did not comply with this decentralized structure. In order to fit the information systems structure to the

organizational structure, the systems in-place needed to be broken down into small pieces that were more manageable from the organizational, operational, and technological point of view.

To achieve this, each business division was given the responsibility to run their own data center, having its own processing capabilities, database, and operational staff. Thus, systems were decentralized and autonomy was granted to the business divisions in all the dimensions, hardware, operational control, software development, data control, and management control. As a result, over 90 computers and more than 20 applications were put in place. (Figure 6 shows the proliferation of computers at that time.) The environment included five processor vendors (among them were, DEC, Prime, and Data General,) six distinct processor families, ten distinct operating systems, eight distinct programming languages and independently developed software modules. This variety of hardware and software had significant implications for the cost-effectiveness and flexibility of the systems.

The information systems people have expressed that "the decentralization of the systems was driven by the organizational structure and a price on technology had to be paid." A price on technology was paid because the trend in computer technology in the 1970's was towards centralization. IBM had very few minicomputers to offer. Instead, IBM offered

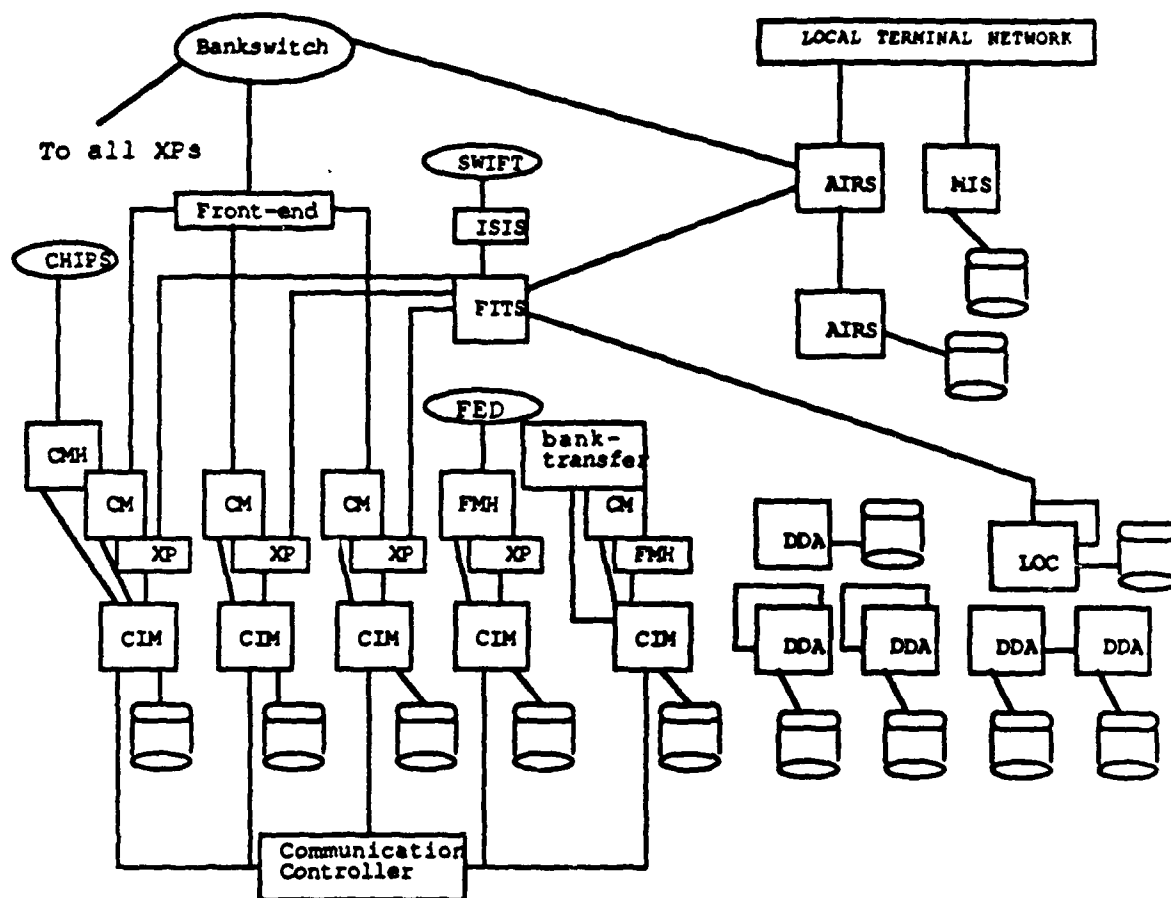


Figure 6

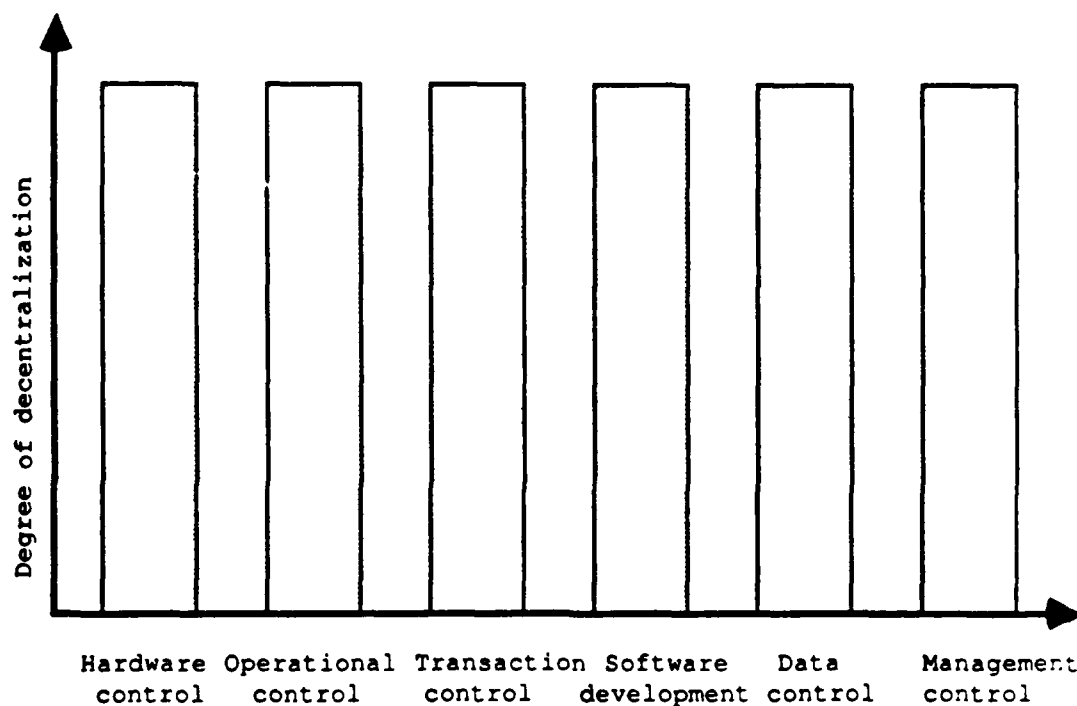
1970's Financial Institution systems

large computers with software off-the-shelf, all utilities incorporated into the system, and communication facilities. On the other hand, DEC provided better minicomputer systems but with almost no software available for their systems. Since the management goal was to give a computer to each business group, minicomputer technology was chosen. So, even though in the 1970's the best choice would have probably been IBM since it provided systems which have more software available off-the-shelf, they could not afford to place 20 IBM computers for the Financial Institution systems. Therefore, in the hardware evaluation process, the technology issue was outweighed by organizational priorities.

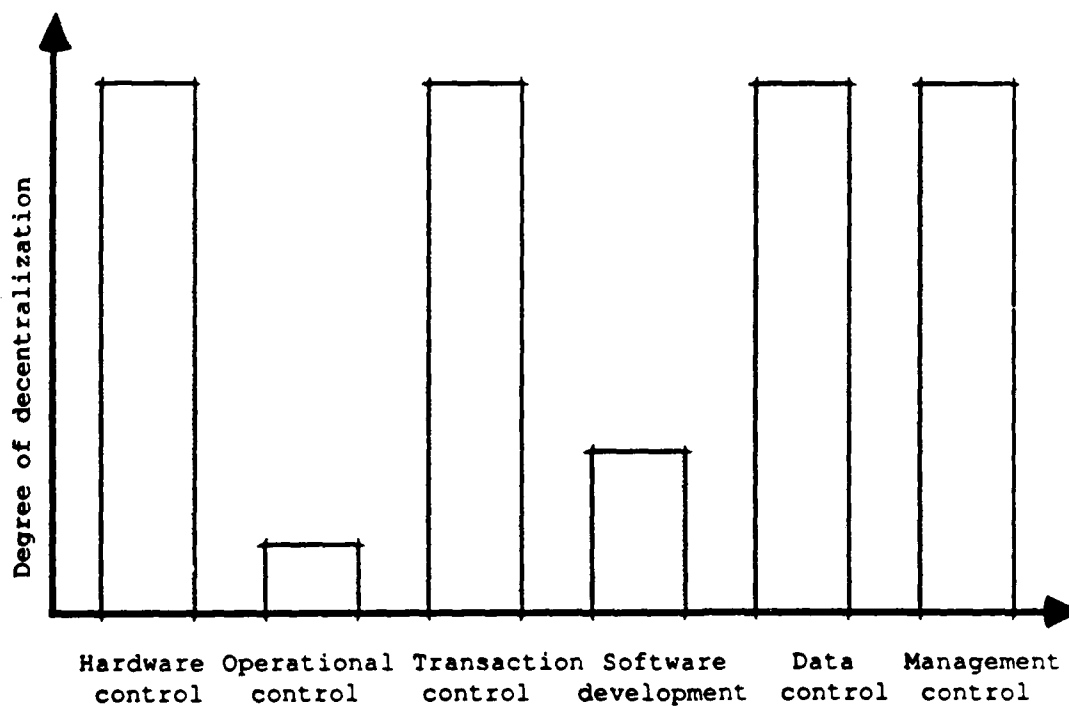
However, some people within the bank have expressed that "even if a high price on technology was paid, it was the right thing to do because technology ought to respond to the business instead of driving the business." In retrospect, the bank by following this philosophy of autonomy and decentralization was relatively successful at completing systems to meet business objectives. In contrast with this bank, other banks that followed a centralized approach were not so successful at completing the same kind of systems. The reason for this was the degree of complexity of the systems they were developing.

The main criticism of this stage is precisely its high degree of decentralization. The systems implemented were too small, too heterogeneous, and too numerous. Most business divisions

did not have the expertise to either acquire hardware for their business needs or to develop the necessary software in-house. Thus, different consulting firms were hired for the development of the software. This was costly and the resulting systems had very little commonality. Furthermore, the hardware configurations were different and in many cases, incompatible to each other. There was a lot of duplication in the development of the systems, and a lot of money and time had to be spent in building interfaces among these heterogeneous systems. Finally, each business divisions had its own data center having the responsibility to run and operate them. The degree of centralization vs. decentralization at this time is shown in the following graph:



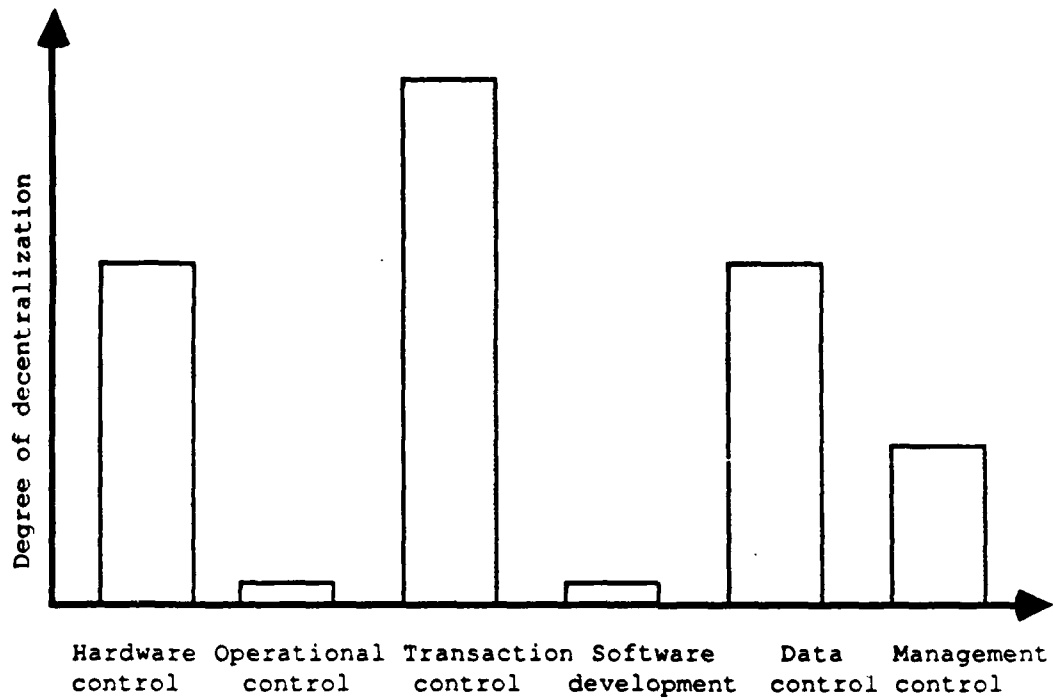
It did not take very long before management realized that in order to give autonomy to the business divisions, they did not have to relinquish the control over the hardware, software development, and operational control of the data centers. The lack of expertise from the business divisions to run and operate the data centers became a major problem, and the operational control of the data centers was given back to the systems group. Thus, each business group had its own data center but the systems group had the responsibility to operate them. On the other hand, the business divisions kept the transaction control, i.e. data entry, repair and verification of their transactions. The degree of recentralization in the 1970's is shown in the following graph:



The diversity of hardware and software, and the high degree of decentralization of the 1970's had two implications for the Financial Institution systems: 1) Integration. There was an increasing need for integration, but the proliferation of heterogeneous systems made it difficult to integrate the systems. 2) Evolution. The systems in place were difficult to evolve to meet new business requirements and to accomodate growth. Thus, in 1982, a new architecture was developed to mediate the conflicts between autonomy and integration. This new architecture which is called the conceptual model would provide supervisory functions, terminal grouping, and security functions as if every business division had its own computer but without having to give one computer to each single division.

So, even though in the 1980's the economics of computer technology tend to favor the distribution of processing power, there are other problems that may work against this trend. Distributed computing implies, operation of the computers, duplication of functions, building of interfaces to communicate them, and creation of new bottlenecks, such as a communication interface. Thus, from the systems perspective, it would be more cost effective to centralize some of the functions. Following this, the bank started a move to change from a physical distribution to a logical distribution. The degree of centralization vs. decentralization in the 1980's is

shown in the following graph:



Chapter 5

The CIS Conceptual Model

The CIS conceptual model was developed as a baseline architecture for the Financial Institution systems seeking technical solutions that do not conflict with the organizational characteristics of the bank. This model is evaluated against the principles of autonomy, integration, and evolution. The model is composed of six components and it is shown in Figure 7 [8].

5.1 CONCEPTUAL MODEL COMPONENTS

1. External Interface

The external interface component is responsible for the interface between the bank's systems and external entities which communicate electronically with the bank. This component is responsible for the communication to:

- Payment networks, such as Fedwire and CHIPS.
- Communication networks, such as SWIFT.
- Other systems within the bank.
- Customer terminals.
- Workstations professionals.

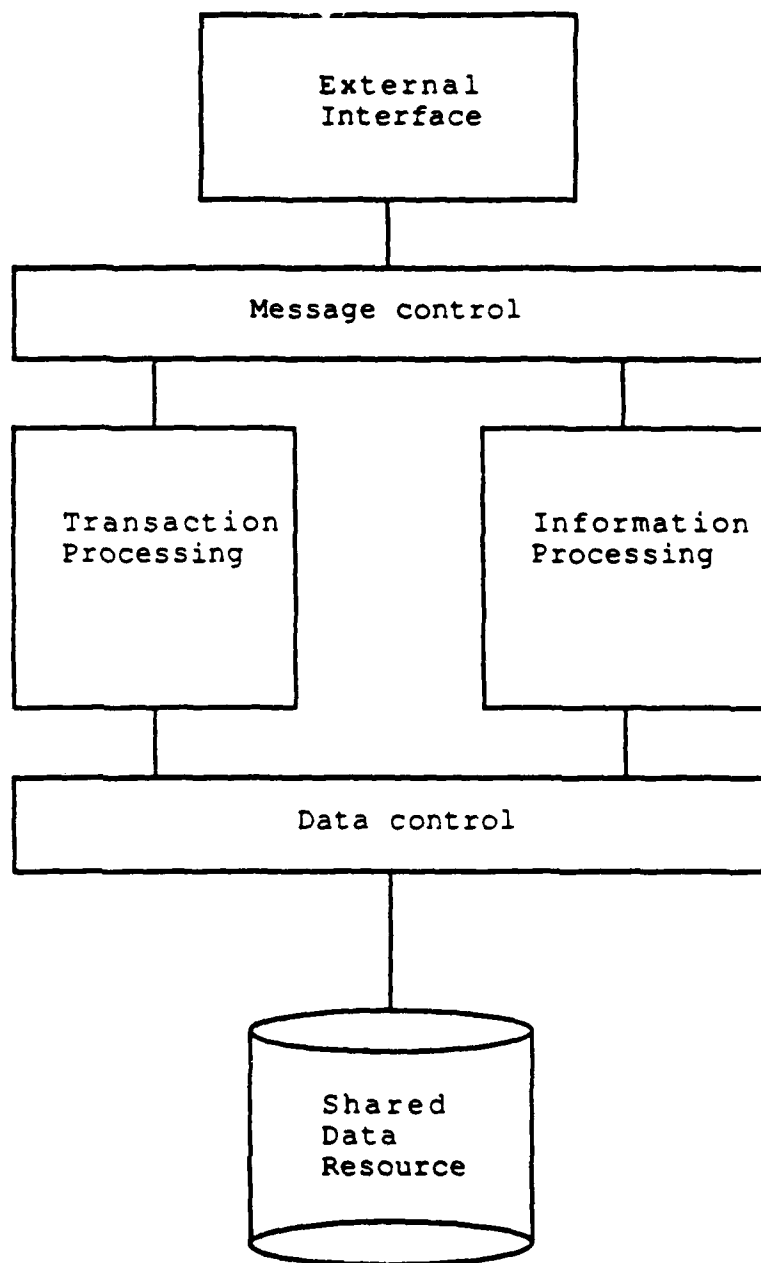


Figure 7

The CIS conceptual model

There are three modes of communication that the External Interface component should support:

1. Message exchange. The exchange of communication is a discrete element which contains the information required to satisfy a service request.
2. Interactive form. The exchange of communication is established by a dialogue between systems. In a dialogue usually more than one message is transmitted.
3. Bulk form. This is similar to the message exchange mode since it is the transmission of complete and discrete information packages. However, the size of bulk packages is generally larger than a message.

2. Message control

The message control component is responsible for the transmission of information between different processing components, i.e. inter-computer communication, terminal control, etc. This component is responsible for performing the following functions: 1) Routing which accepts a request and determines the physical address to which the message should be delivered to. 2) Translation which maps a limited number of protocols from one standard to another. 3) Sequencing which determines the order to deliver messages. And 4) Monitoring which determines the integrity of messages as they are transmitted through the system at any given time.

3. Transaction Processing

The transaction processing component is composed of all the applications which result in changes to the financial condition of a customer. The ultimate result will be updates to the database. This component performs the following functions:

- Risk management functions, such as checking that a transaction does not violate any condition imposed by the bank, customer, or regulatory agency.
- Validation functions to ensure that all information required for processing is provided, and if provided that it is correct. The validation and repair of transactions will specifically be designed for the application.
- Accounting. The TP component should record all transactions being performed for accounting purposes.
- Management Information Recording. The TP should also record the necessary information to support the decision-making processes of the bank.

4. Information Processing

The information processing component is composed of all the applications that provide information to support the decision making of both customers and internal management; no changes occur in the financial condition of a customer. This component will communicate with databases which are static and historic in nature. This component should perform the

following functions:

- Data analysis to provide the capability to present conclusions based on the data analyzed. The source of this data can be both internal and external to the bank.
- Ad Hoc reporting to provide the capability of obtaining reports as they are needed. The information is based on transaction processing functions or external sources.
- Static reporting. The production of reports which have been defined in advanced. The data to generate these reports is usually obtained from the Transaction Processing component or from external sources.

5) Data control

This component controls the access from the processing components to the information that is common to several applications. The following functions are performed by data control: 1) Concurrency control which ensures that multiple requests do not change the same piece of data at the same time, thus preventing inconsistency of data. 2) Update propagation which makes sure that redundant data is updated appropriately. 3) Security which prevents unauthorized access to the database, and controls the view of the data permitted to different users. 4) Routing which determines the segments of the database to be accessed in response to a request, and returns its result to the appropriate processing element. And 5) Sequencing which determines the order in which requests

AD-A195 856

INTEGRATING INFORMATION SYSTEMS IN A MAJOR
DECENTRALIZED INTERNATIONAL OR. (U) MASSACHUSETTS INST
OF TECH CAMBRIDGE A GUPTA ET AL. DEC 87 MIT-KBIISE-7

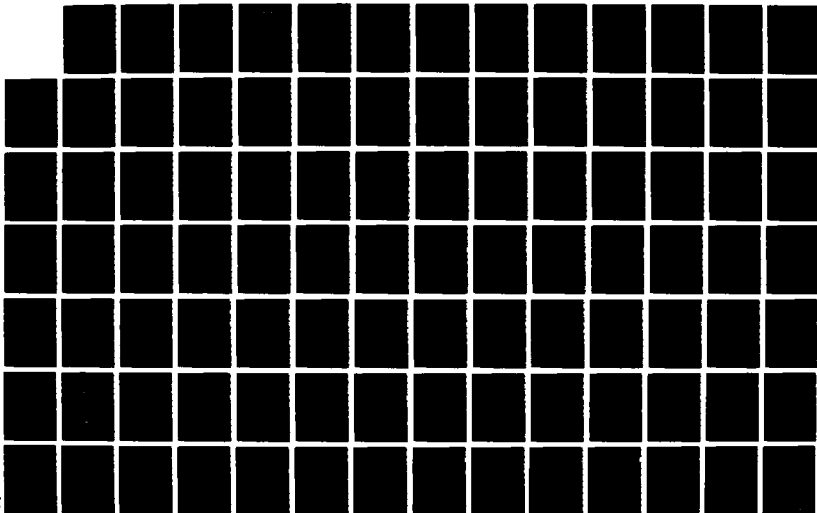
3/3

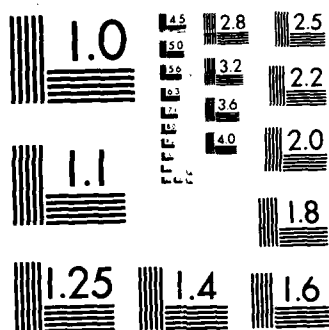
UNCLASSIFIED

DTR557-85-C-00083

F/G 12/7

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

should have access to the database.

6) Shared Data Resource

The shared data resource component is responsible to maintain all the information that is common to different components of the systems. This component does not have to be a single database but it could be distributed according to the needs of information that different processing components have.

This model allows for the inter-organization and intra-organizational potential applications that Madnick and Wang have identified [2]. For example, the Message control component allows for inter-computer communication, so even if different applications are hosted in different computers, transactions can be sent through the centralized message control. Furthermore, it provides a single gateway entrance to all applications which permits communication to any application from any terminal. For example, data can be collected and entered from a single point and then distributed to all the applications requiring that particular information.

5.2 AUTONOMY, INTEGRATION, AND EVOLUTION ASPECTS OF THE MODEL

An ideal implementation of the conceptual model for the Financial Institution systems is presented in figure 8.

5.2.1 AUTONOMY.

The autonomy principle is reflected in the conceptual model by providing the ability to develop systems independently, i.e., the development of one application is not dependent on the development of any other application. In fact, each one of these applications can be hosted in a different computer. In the CIS conceptual model, the components are independent of each other with the exceptions of, the data control, and the message control components. The applications contained within each component can be as autonomous as desired. In fact, they can have their own hardware, software, and database.

The autonomy in the development of systems can lead to independent systems having autonomous databases which is in conflict with the shared data resource component of the CIS conceptual model. If the independent applications have common data but they do not share the database, the result will be to have "shadow databases" of the original database (i.e., redundancy). One of the reasons for different systems to have independent databases is that managers may not be willing to wait upon the completion of a global database that will be shared by several applications in order to start the development of their own systems.

The infrastructure outlined in the conceptual model does not interfere with the autonomy of the divisions to have their own

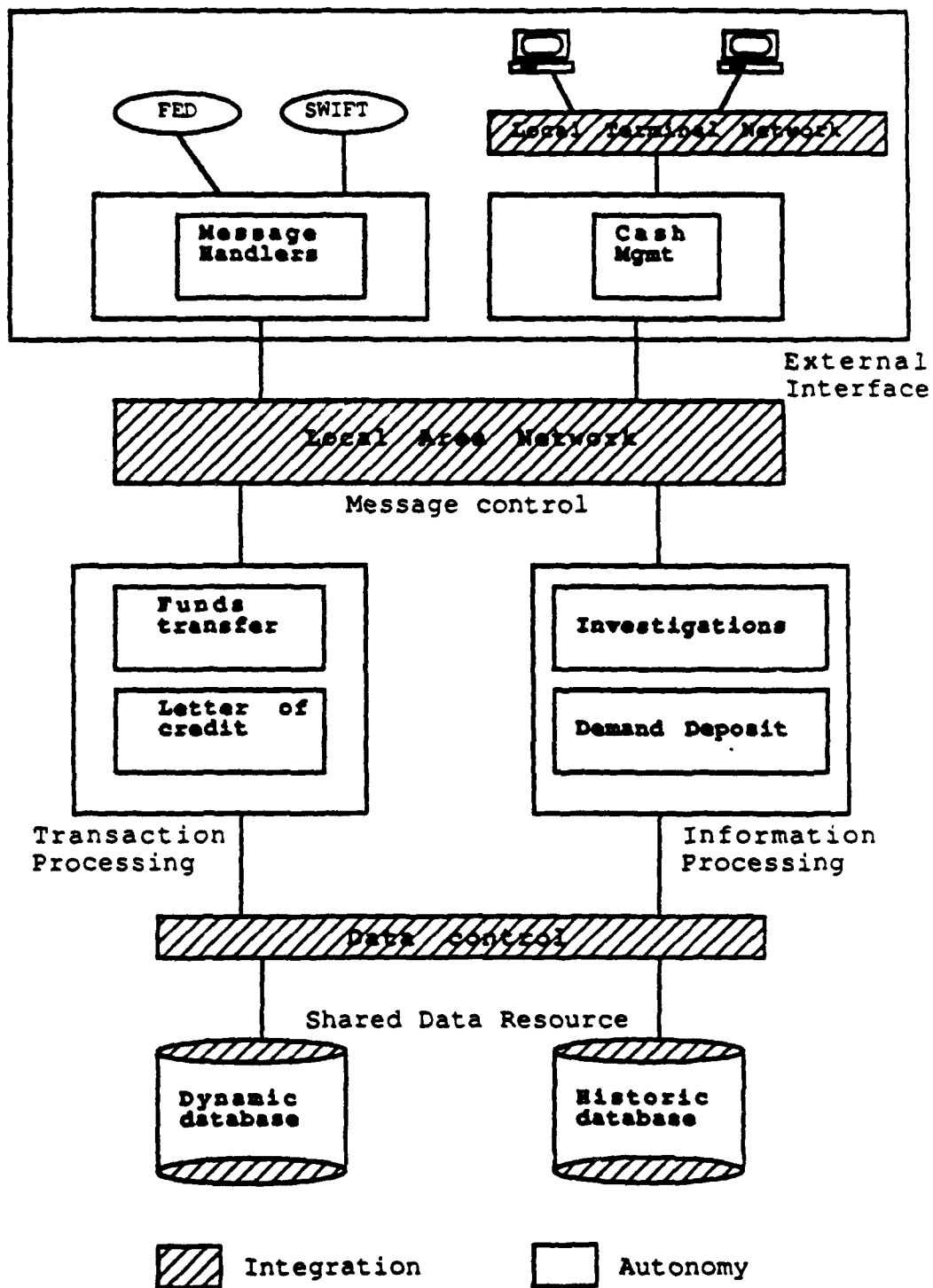


Figure 8

Ideal Implementation of conceptual model

operators for data entry, repair, and verification. Thus they still have control over their own transactions. Even though all the transactions may reside in the same computer, and they share the same transaction queue, the system will behave as if they were independent systems.

5.2.2 INTEGRATION.

The conceptual model also conforms to the integration goal. There are two components that are critical for integration, the message control, and the data control. The message control component enables inter-computer communications. Thus, even if applications are hosted in different computers, the protocols to communicate among them have been standardized facilitating their interfacing. Moreover, this component provides a single entrance to all applications for Financial Institutions. Thus, any terminal will have the capability to access any application. This makes it possible, the single customer interface for all the products and services offered.

The data control allows for the control and sharing of information that is common to more than one application. Different applications can have different views of the same share data resource. Therefore, functions such as, risk management control that require the communication and integration of several application can now be performed.

5.2.3 EVOLUTION.

Finally, the conceptual model provides a much better environment for evolution, in terms of volume growth and creating new products and services since it provides for better integration among applications, and its modular characteristics allows the extension of hardware to process more transactions.

Chapter 6

Analysis of the Financial Institution Systems

In this chapter, the current implementation of the Financial Institution systems is analyzed. This implementation follows the guidelines of the conceptual model described in chapter 5. In particular, this chapter contains an analysis of the following components of the Financial Institution Systems, the Funds Transfer system (FTS), the Demand Deposit Account system (DDA), and the Transaction Investigation system. No analysis is provided for the Cash Management System (CM) since it was only rehosted from a Data General to a DEC VAX with no further changes.

TECHNOLOGY CHOSEN

The technology used has been standardized to DEC hardware, and to the MVS operating system.

1. VENDOR

The vendor chosen was DEC. The main reasons for this selection were that the information systems group has had extensive expertise with DEC systems; and that most of the Financial Institution systems were already running on DEC systems (PDP 11). This reduced the complexities and risks of transition to the current implementation.

2. PROCESSOR

The specific processors chosen were the VAX 785 and the VAX 8600. The reasons for this selection were:

- Performance: The VAX/785 has been rated at 1.5 MIPS and the VAX/8600 has been rated at 4-5 MIPS. The VAX/8600 represents an improvement of over 500% in price performance over the previously used PDP 11. MIPS are an important measure since most processors were running up to capacity and some of the systems are CPU intensive.
- Compatibility: The VAX 785 and VAX 8600 are totally compatible with the entire family of DEC processors and peripherals.
- The VAX 785 and VAX 8600 can enter a VAX cluster. DEC VAX cluster architecture allows several VAX computers to share multiple sets of data files.

3. DATA COMMUNICATIONS

The communications technology chosen was Digital Network Architecture using Decnet/Ethernet which are established products for communication between DEC computers. The physical capacity of Ethernet is 10 million bits per second which exceeds the current needs of FI systems.

4. SOFTWARE

The software chosen was DEC's VMS operating systems. VMS contains all the basic systems services and utilities required to develop the different applications. However, different database technology and programming languages were still chosen for the different applications.

6.1 ANALYSIS OF THE FUNDS TRANSFER SYSTEM.

To analyze the Funds Transfer System in perspective, a brief evaluation of the system under the previous architecture is discussed. Then, a description and analysis of the current implementation is done.

6.1.1 THE FUNDS TRANSFER SYSTEM UNDER THE 1970s ARCHITECTURE

The Funds Transfer System set up in the 1970s is illustrated in figure 9 and presents the following characteristics [9]:

1. Customers were classified by banking groups (e.g. North America, International, etc.) Each banking group had an autonomous processing environment, with its own processor capability to host the application program namely, Customer Information Manager (CIM). Each group also had its own front-end and external interface hardware and software, and its own independent database. There were a total of five of these environments. One vendor was contracted to develop

the CIMs software, whereas two other vendors developed the front-end, and external interfaces.

2. The hardware chosen was minicomputers. Each CIM was hosted in a minicomputer, and each external interface and front-end were hosted in another minicomputer. Most minicomputers used were PDP 11.
3. The CIMs and front-end applications were mainly developed using low level languages (assembly language), in contrast to using high level languages that would be compatible among different hardware configurations.
4. Communications among CIMs was necessary since most funds transfer transactions generated secondary transactions to be processed in other CIMs. A Communications Controller was in place to communicate transactions/messages between the CIMs.

The problems this architecture posed were numerous. For example:

1. When customer groups were reorganized as the organization of the bank changed, this architecture made it difficult to accomodate needed organizational changes since each banking group had its own processor and database.
2. On average, each FTS transaction spawned 2 1/2 secondary

FUNDS TRANSFER SYSTEM
OLD ARCHITECTURE

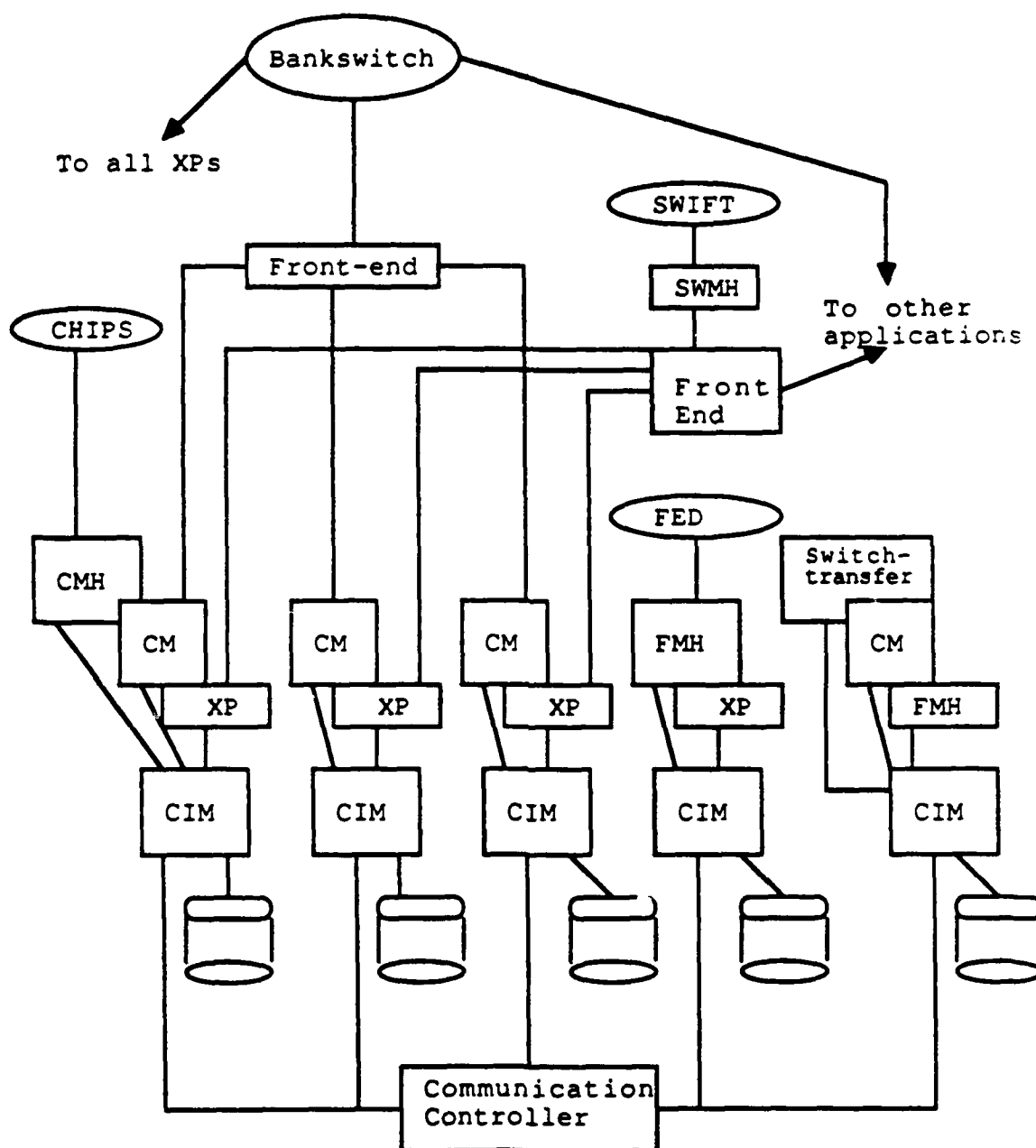


Figure 9

FTS systems - 1970s architecture

transactions to be processed in other CIMs [9]. The fact that CIMs had independent databases increased dramatically the number of transactions that needed to be processed. For example, if 20,000 funds transfer operations were generated daily, an average of 50,000 transactions would be processed.

3. A new bottleneck was created in the system at the Communications Controller since the transmission of transactions among CIMs increased dramatically as volume increased.
4. Each banking group had its own processor for the application (CIM), and for the external interfaces. This created a proliferation of minicomputers, heterogeneous software, duplication of functions, and difficulty in integrating the Funds Transfer System application to other Financial Institution Systems.
5. Given the increasing demand for funds transfer operations, the Funds Transfer System was nearing its maximum capacity, thus degrading to unacceptable limits for customer service the processing capabilities of FTS.
6. The PDP 11, introduced in 1970, was reaching the end of its product life cycle. New hardware was providing much more capacity in terms of MIPS, (processing capacity), and better interfaces to newer peripheral devices.

The main technical problems preventing the easy integration of Funds Transfer applications in the old implementation were, the one to one relation between minicomputers and applications, heterogenous software, the saturation of processing and network capacity, and the independent databases per CIMs. The autonomous and numerous data centers resulted in duplicity of functions which had as a consequence high operational costs.

6.1.2 THE FUNDS TRANSFER SYSTEM UNDER THE NEW ARCHITECTURE

As a result of the limitations posed by the previous architecture, a new funds transfer system was designed in 1984 following the conceptual model guidelines to achieve the required level of integration, processing efficiency, and control. Five main decisions were made:

- A homogeneous operating environment was to be created.
- A unified database was to be used using the DEC VAX cluster technology.
- A centralized and high performance network for inter-computer communications and terminal support was implemented.
- Centralization of message and payment interfaces support was implemented.

In the new Funds Transfer system, there were fundamental changes in the architecture of the system but no reinvention or redesign of the applications was made. The new system (FTS) was divided into four main components, shown in figure 10. These four components are, the Message Processor (MP), the Transaction Processor (TP), the Exception Processor (XP), and a unified FT database.

The specific processor chosen to host the applications was the VAX 8600. In particular, 4 VAXs 8600 were used, one for the Transaction Processor, one for the Exception Processor, and two for the Message Processor. Furthermore, FTS was implemented in a VAX cluster, so that the processors for TP, XP, and MP could share multiple sets of data files. The VMS operating system and the programming language C were chosen to develop the applications. The programming language C replaced the low level languages that were used previously to develop applications.

A description of the four components of FTS follows:

1. Message Processor (MP).

The Message Processor has responsibility for receiving and transmitting messages between the Funds Transfer applications and all the external entities. The external entities are among others; the New York CHIPS, the Federal Reserve Bank, SWIFT, and an Electronic Banking Micro located in the

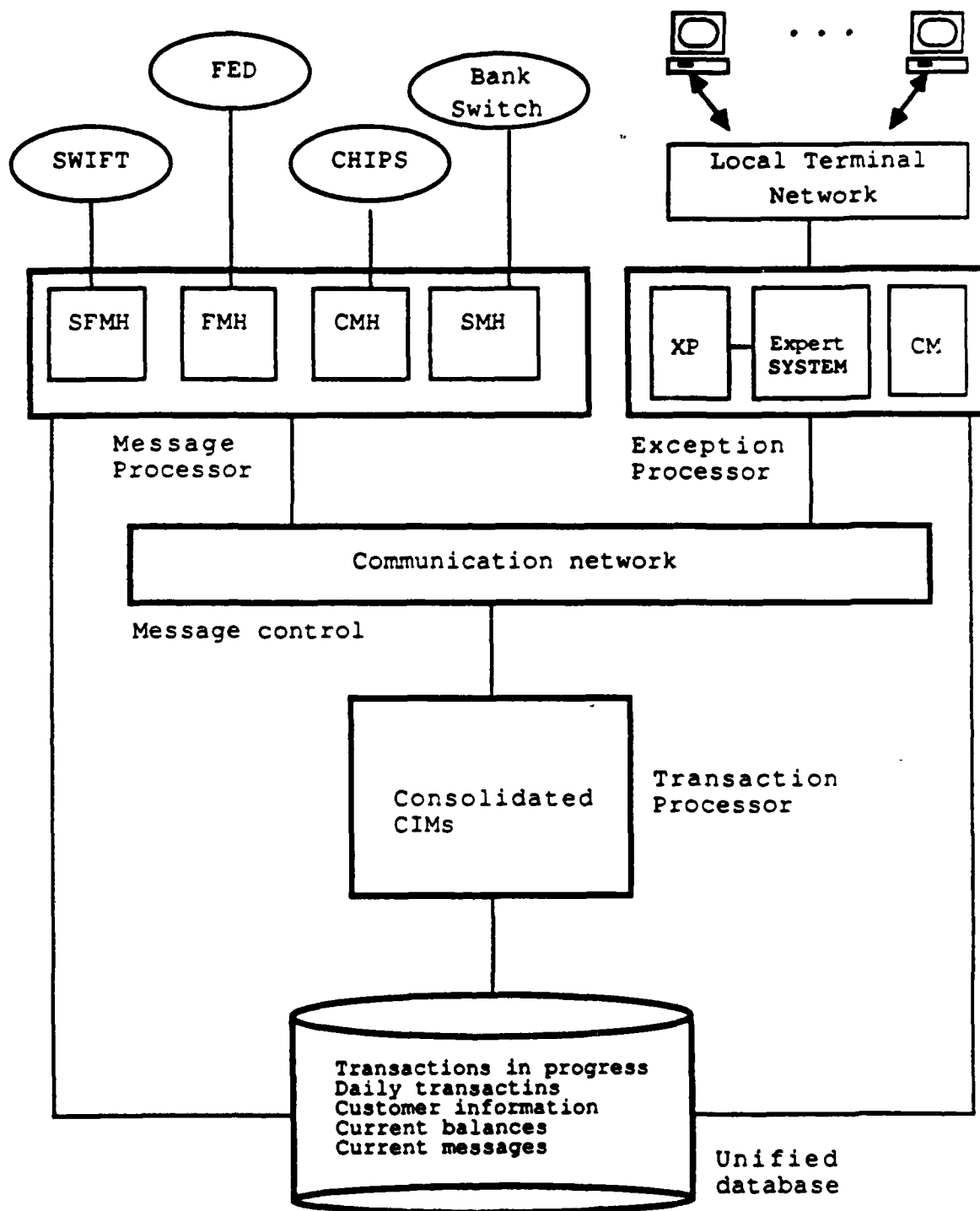


Figure 10

Funds Transfer System - New architecture

customer's offices. One VAX processor is used to host all the payment network interfaces (e.g. FED, CHIPS). Another VAX processor is used to host all the message network interfaces (e.g. SWIFT). Thus, the message processor consolidates all the external interfaces that were previously integrated into the application and running in different hardware as illustrated in figure 9. The functions performed by the MP for all the messages types include, test/authentication, addressing, editing of message content, and reformatting of messages. If the message received is unstructured, the information provided is incomplete, or any errors are found, the message will be routed to the Exception Processor.

2. Transaction Processor (TP).

The Transaction Processor replaces and performs all the functions previously assigned to the CIMS and the Network Controller. Its functions include, balance and amount controls, debit authority controls, and posting. The message is routed to the Exception Processor if any errors are found, or any decision needs to be made.

3. The Exception Processor (XP).

This application handles all the exceptions: 1) Errors discovered during the Message Processing or Transaction Processing. 2) Unformatted messages or messages with incomplete information. 3) Transactions that require decision making capabilities. In these cases, the message is routed to

the Exception Processor. The XP supports the following functions: data entry, repair of messages, and database maintenance and inquiry functions. An expert system has been developed to support some of the functions performed by the Exception Processor. This expert system attempts to interpret unstructured or incomplete messages. If the expert system can perform the interpretation, an operator only has to check the expert system output against the original message. If the expert system cannot interpret the message, the message is then manually interpreted and reformatted by an operator.

4. A Unified Database.

A unified database was designed to be shared by the Transaction Processor, the Message Processor, and the Exception Processor. The unified database substitutes the segregated CIM databases in the previous architecture. The Funds Transfer system manages over 30 files which are classified into transaction files, static files, and dynamic files. An average of 40 to 50 physical accesses to disk are made by each funds transfer transaction. The physical accesses include, get customer information, message repair, get/change customer balances, communication among systems, and recording of messages for restore and recover capabilities. Considering that an average of 60,000 funds transfer transactions are processed daily, the number of accesses to disk are significant.

The software used to manage the FTS files are, TMX-32 software, and the DEC RMS file system. The TMX-32 software is a transaction manager which is used to manage the transaction queues. The main reason for using TMX-32 is the need for high performance, recoverability, and ease of maintenance. On the other hand, the VAX RMS file system is used to manage the dynamic files. These files contain information such as, account balances, and overdraft authorizations. Finally, the static information files which contain information such as, customer names and account numbers are read directly from within the memory for performance reasons.

TRANSITION PLAN

The entire plan would take three years to be implemented. A transition plan was devised to rehost all the applications in the new hardware. The plan was divided into eight independent projects with their own project manager and staff. The eight projects included the rehosting and design of the new Transaction Processor, the Exception Processor, and the Message Processor. Although there were fundamental changes in the architecture of the system, no reinvention or redesign of the applications was considered except for integrating some of the segregated applications. Finally, functionality has been added incrementally to the expert system to support the reformatting of unstructured messages coming from the FED, CHIPS, SWIFT, and all the other message handlers.

6.1.3 A FUNDS TRANSFER TRANSACTION EXAMPLE

A typical example of a request for a funds transfer from say, Company A in Boston to Company B in London going through the Funds Transfer system is presented below. Here, company A in Boston wants to make a payment of \$35 million to Company B in London which has a dollar account with the bank's London branch. A flowchart showing the sequence of steps followed by the transaction is shown in figure 11 and explained below:

1. Company A has an account in a bank in Boston and calls its account representative to perform the payment. The bank in Boston transfers the amount from Company A's account to its own account.
2. The bank in Boston performs a payment to the FED.
3. The FED in NY performs a payment to our bank in NY.
4. At this point, the bank's Funds Transfer system receives a payment message from the FED. Figure 12 shows how this funds transfer transaction flows through the bank's Funds Transfer system. The payment message is received by the Message Processor and is read by the FED Message Handler (FMH) that is hosted in the Message Processor. The FMH acknowledges the reception of a message by recording a copy of the message in the database. After the FMH has read the

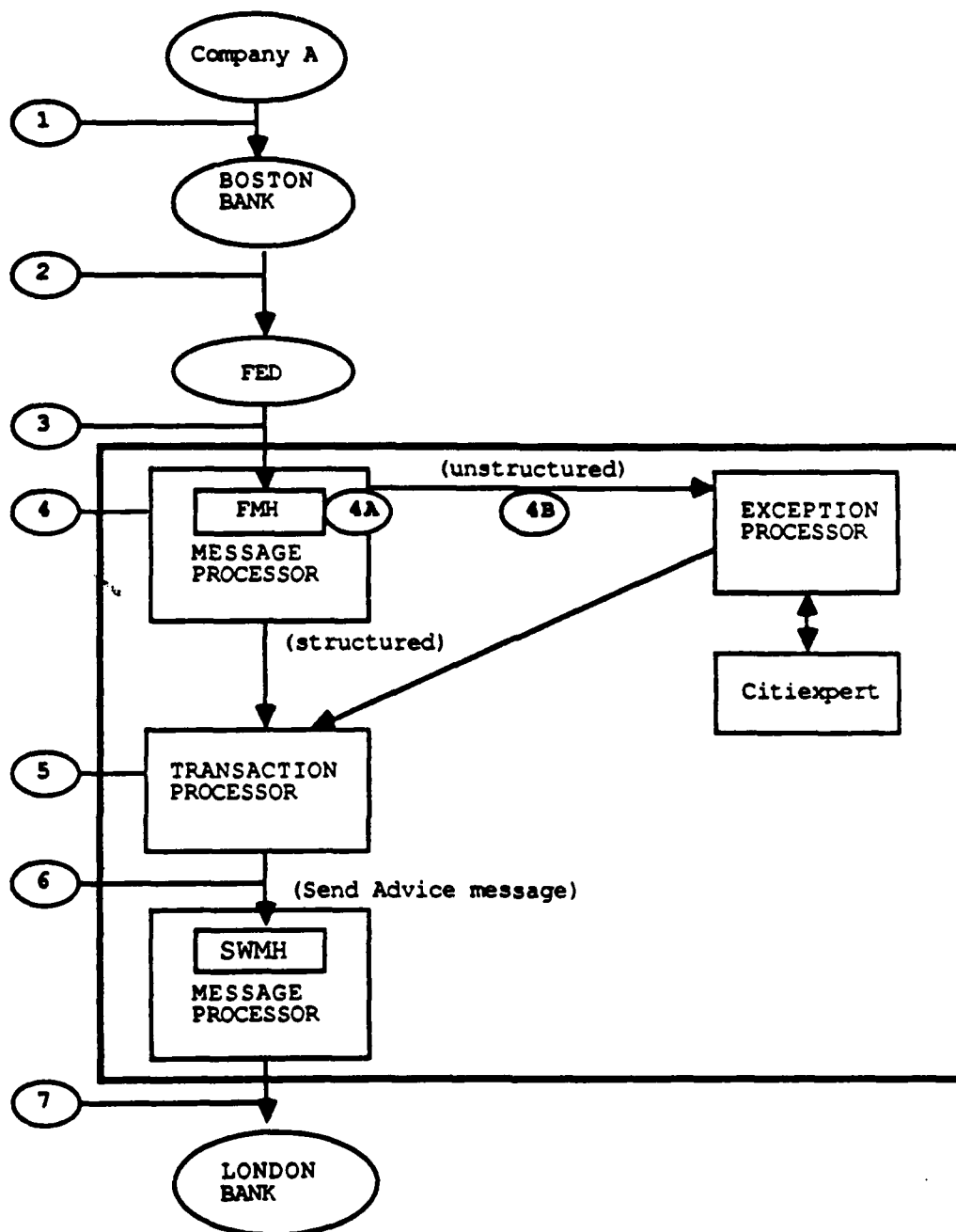


Figure 11

Funds transfer operation

message, it attempts to interpret it. Two different paths are followed depending on whether the message is structured or unstructured:

A. Structured message. FMH interprets the message by extracting information such as, the originator, the destination, the amount of the payment, and some other reference information. A Financial Transaction record (FTR) is created with this information. Other information such as, the name and address of the destination entity, and the day and time at which the message was received is also included in the FTR. The FTR is then recorded in the database. An average of 75% of the messages are structured and go through the system with no human intervention.

B. Unstructured message. FMH can not interpret the message because some information is missing, the information in the message cannot be extracted, or some decision making is needed. Thus, this message is sent to the Exception Processor through the Local Area Network for repair. The Exception Processor is supported by an expert system, which interprets the message, repairs it, and creates a Financial Transaction Record. If the expert system can interpret the message, an operator will verify that the expert system has created the FTR correctly. If the expert system cannot interpret the message, it will send

it to the Exception Processor to be repaired manually by an operator.

5. Once the message has been interpreted and the Financial Transaction Record has been created, the FTR is routed to the Transaction Processor for processing. The destination, the London branch of the bank, has an account in US dollars in NY, and is credited with the \$35 million. Before crediting an account, the Transaction Processor performs other operations such as, check for availability of funds, check authority to debit account, etc. This checking is done by the Transaction Processor against the database that contains all the information. When the Transaction Processor is done, it creates an accounting record that is written in the database.
6. After the transaction has been processed by TP, (i.e. the money has been credited to the London account), a message is sent through SWIFT to the London branch of the bank indicating that its account in NY has been credited with \$35 million to be paid to Company B. The SWIFT interface application which belongs to MP, acknowledges the reception of the message by recording it in the database.
7. The bank in London receives the message advising for the credit in its account and notifies Company B about the payment.

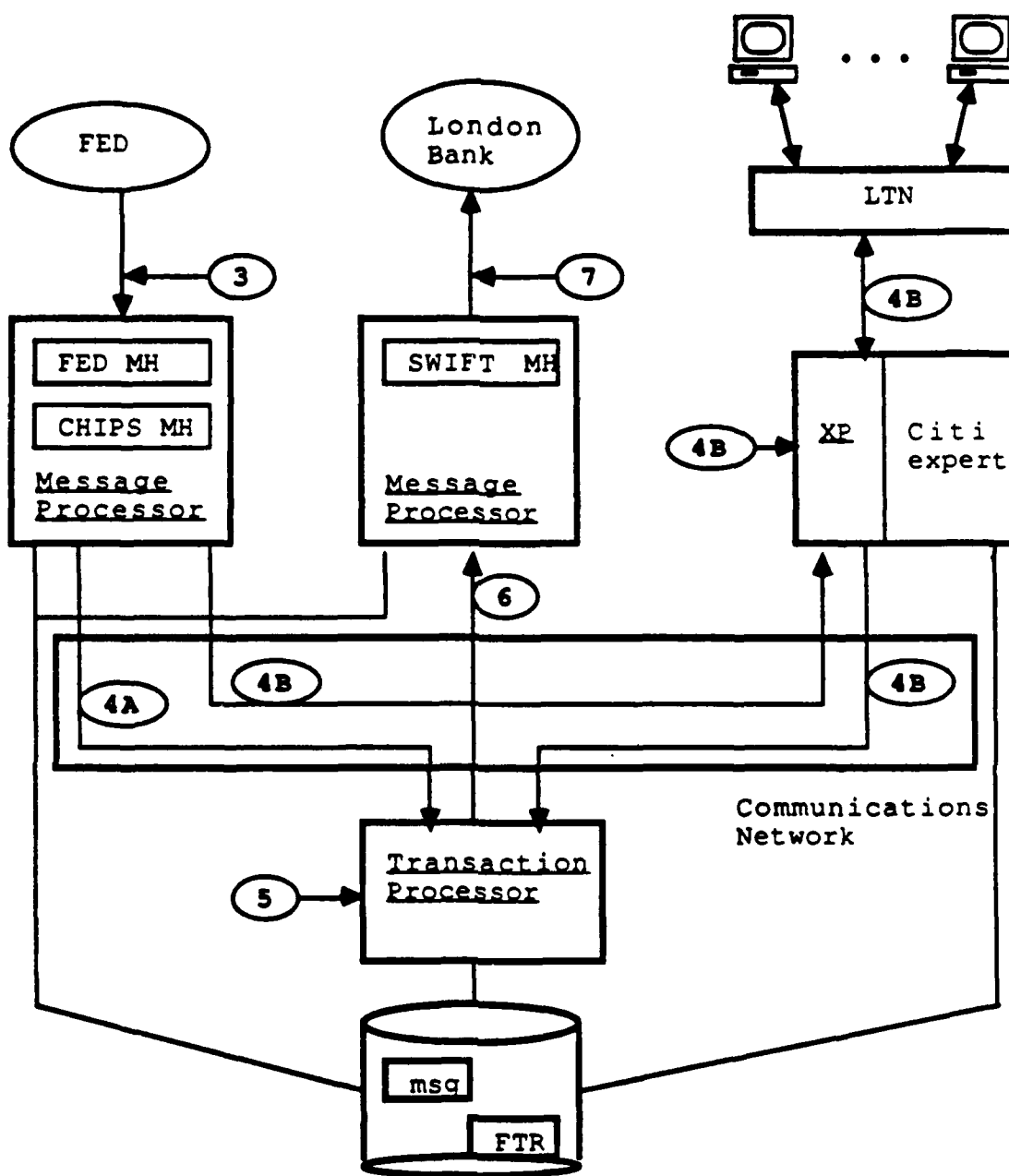


Figure 12

Funds Transfer transaction through FTS system

6.2 ANALYSIS OF THE DEMAND DEPOSIT ACCOUNT SYSTEM

An analysis of another component of the Financial Institution Systems, the Demand Deposit Account System (DDA) follows [10]. Before evaluating the current implementation of DDA, a brief evaluation of the system under the previous architecture is made.

6.2.1 THE DEMAND DEPOSIT ACCOUNT SYSTEM UNDER THE 1970s ARCHITECTURE

The Demand Deposit Account System in the 1970's is illustrated in figure 13 and presents the following characteristics [10]:

1. The DDA system is divided into three applications, STAAR II, STAAR DFI, and STAAR. STAAR handles the overseas branches, STAAR II handles the International Financial Institutions, and STAAR DFI handles the Domestic Correspondent Banks. STAAR II and STAAR DFI are functionally the same. However, STAAR DFI was added to DDA because STAAR II did not have the capacity to handle the additional volume.
2. The hardware chosen were minicomputers. Most minicomputers were PDP 11s. Each DDA application was hosted in a minicomputer making a total of seven minicomputers to host all the DDA applications.

3. The DDA applications were developed using the operating environment RSTS/E.
4. Applications used non-standard protocols for communications.

The problems this architecture posed were numerous:

1. The DDA systems were often unable to finish the processing on time causing delays in delivering information to customers (e.g., to provide daily balance updates), and to other systems, (e.g., to Funds Transfer System). The activities that posed the greatest problem were: 1) Delays in tape hands-off to Cash Management, and Funds Transfer; 2) Delays in electronic transmissions to SWIFT and the bank switch; and 3) Inability to provide on-line availability for more than 12 hours since the processor was used for intense batch processing.
2. The system also presented functional bottlenecks. The system did not allow for: 1) Adding additional on-line links which would have brought transactions into the system at an earlier time. 2) Adding additional on-line functions which would have allowed more efficient inquiries. 3) Adding additional off-line functions which would have improved the report generation. These limitations were caused by the hardware and software environment. For example, the PDP 11 limited the maximum allowable program size making it

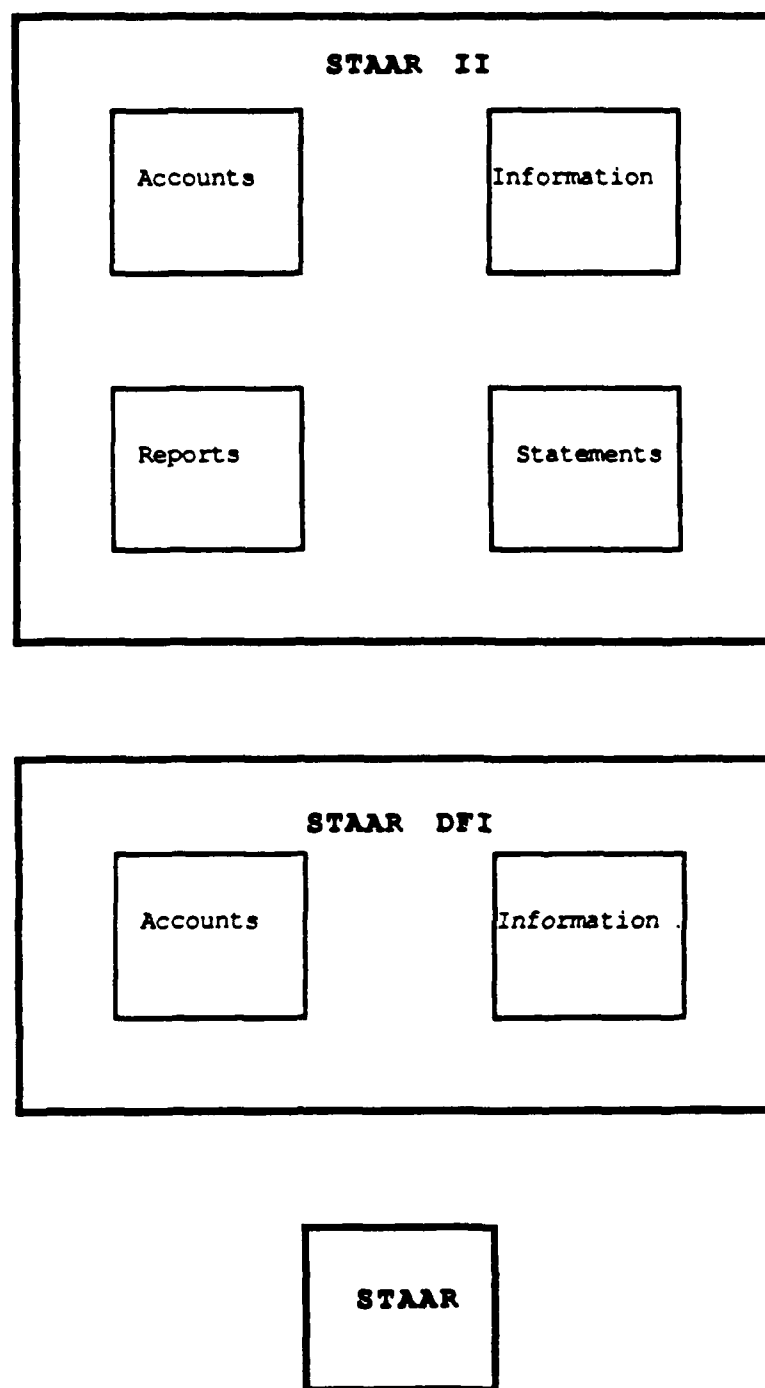


Figure 13

DDA system - 1970s architecture

difficult to add functionality to individual on-line and off-line programs. Furthermore, the limited processor capacity of the PDP 11 (0.5 MIPS), made it very difficult to add functionality since the processors were used up to capacity. Finally, the limitations of the operating environment RSTS/E, made it difficult to add functions to the on-line systems.

3. The DDA applications ran in three sites and involved seven machines. These sites were in different floors causing the duplication of functions which translated into inefficient use of resources, increasing operational costs, and increasing costs for maintenance of hardware and software.
4. It was very difficult to integrate the DDA application with the Funds Transfer system in real time since neither the PDP/11 nor RSTS/E could enter a VAX cluster and thus no data sharing could occur.

6.2.2 THE NEW ARCHITECTURE FOR DEMAND DEPOSIT ACCOUNT SYSTEM

In the new DDA system [10], there were changes in the architecture of the system but no reinvention or redesign of the applications was made. The new Demand Deposit Account System development was divided into five stages. In the first stage, the STAAR II and STAAR DFI were integrated into one

system and converted to the VAX architecture. The on-line processing capability of DDA was transported to a VAX/785, and the off-line processing capability was transported to the VAX 8600 (see figure 14). These two processors are more powerful than the PDP/11 in terms of MIPS. This is an important measure since most DDA applications are batch, and CPU intensive. Moreover, although DDA programs have reached their maximum allocable size in the PDP/11, they can grow as large as needed in the VAX processors. The software chosen was DEC's VMS operating systems. The services offered by the VMS operating system are more powerful than those offered by RSTS/E making it easier to add more functionality when needed. The applications were to migrate using the same programming language, BASIC. In the second stage, optimization of the off-line part of the system was to be done if necessary.

The last three stages are long term. Stage 3 will provide integration of the STAAR system to the new DDA architecture. Stage 4 will provide integration with the Funds Transfer System by integrating the on-line segment of the DDA system with the Funds Transfer System. This implies moving posting and account balances to the Transaction Processing function of FTS; and moving the communication interfaces to other banking groups to the FTS Message Processor. Finally, stage 5 will consist on moving the historical inquiries and batch processing to the Information Processing environment.

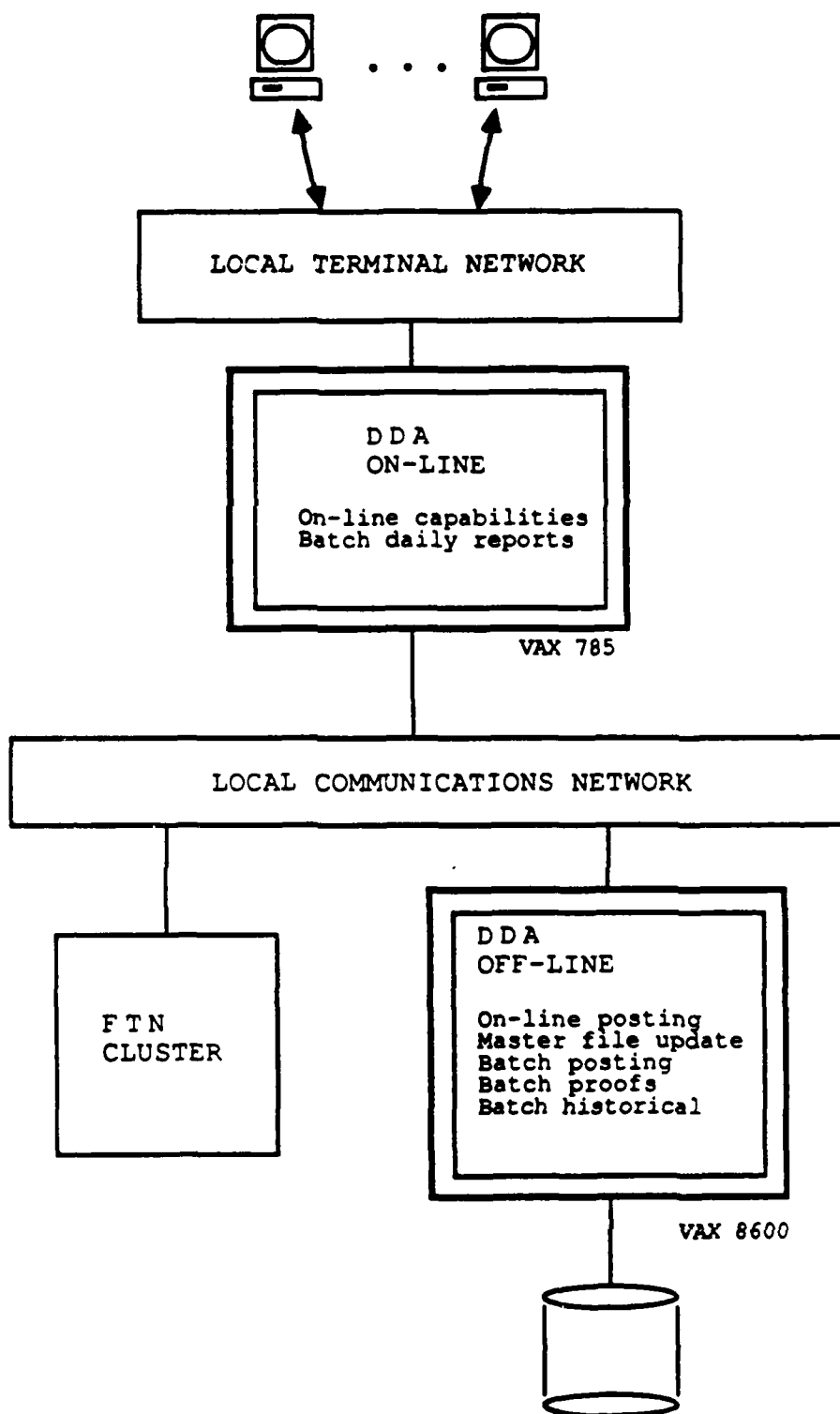


Figure 14

The DDA system under the new architecture

First two stages

The operational and functional advantages of the DDA as a result of implementing the first two stages (i.e., integrating and rehosting the STAAR II and STAAR DFI applications into VAX processors) are:

1. Batch processing runs 30% to 60% faster.
2. Applications are able to handle substantially larger volumes.
3. It is easier to add programs and functions to both the on-line and off-line systems.
4. Two VAXes replace 7 PDP/11s.

Therefore, the following advantages are obtained:

1. Electronic and tape hands-off communication to other systems will be delivered earlier.
2. On-line inquiries will be available 20 hours per day.
3. The monthly statements will be available earlier.
4. Reduction in hardware maintenance costs, physical plant requirements, and operations support.

These functional and operational advantages are derived from using more powerful processors and a more flexible operating environment, and from integrating several applications into less hardware.

6.3 ANALYSIS OF THE TRANSACTION INVESTIGATION SYSTEM

The new implementation of the Transaction Investigation System involved three major tasks [8]. First, the system was rehosted into a VAX cluster using the VMS operating system. A VAX 8600 was used to support the database, and two VAXes 11/785 were used for the application programs. Second, the ORACLE Data Base Management System was used to manage the database namely, the historical database (HDB). The HDB which is a 20 gigabyte database, contains a history of the funds transfer transactions for the previous 90 days and holds approximately 40 million records. Moreover, the investigation itself also generates a history that is also kept in the historical database. Third, the communication facilities for this system were standardized to using DECNET/ETHERNET. The system was designed to support 160 simultaneous users having read-only access to the historical database with an expected response time of 5 to 7 seconds.

The study for evaluating the database management system to be used for implementing the historical database was done in 2 months. This is a typical approach within the bank given its philosophy for obtaining immediate results.

Some problems were found during the development of the system which could have been predicted if an up front study had been carried out. For instance, it was discovered at the system

development stage that the ORACLE "money" data type was not big enough to hold the amounts needed by the bank. Also, that the recovery and re-indexing of the database, which was very large, was taking too long. Therefore, tables had to be partitioned by dates, and applications had to be written to access the database across ranges of dates.

6.4 EVALUATION OF THE FINANCIAL INSTITUTION SYSTEMS USING THE CIS CONCEPTUAL MODEL

The Funds Transfer, Demand Deposit Account, and Investigation systems are evaluated from the integration, autonomy, and evolution perspective. The integration capabilities of these systems (i.e., interaction among them) will be evaluated further in chapter 8.

A. INTEGRATION

In terms of integration, the current implementation of FI systems presents several concurrences with the conceptual model. Some of these represent big improvements compared to the previous architecture:

1. The hardware and operating environment across all systems was standardized. That is, the use of DEC VAXes and VMS operating system.

2. The Communication network (DECNET/ETHERNET) match the Message Control component of the conceptual model, and it allows inter-computer communications. The communication network enables transmission of transactions among the different Funds Transfer components, namely the Message Processor, Transaction Processor, and Exception processor. It also allows communications among the different Financial Institution systems. Furthermore, the communication network represents a single gateway for all applications, thus providing the capability to connect to any application from any terminal.
3. The Message Processor of the Funds Transfer system matches the External Interface component of the conceptual model. All the interfaces to the payment networks are hosted in one processor and the interfaces to message networks are hosted in another processor. This provides several advantages. There is no duplication of interfaces. The external interface is independent of the applications. And, it provides a single entrance for all external entities to the Funds Transfer systems.
4. The Exception Processor component of the Funds Transfer system that used to be segregated by applications is now shared by all the banking groups. It also has access to the unified funds transfer database. An expert system that supports the Exception Processor functions has functionality

to format messages received from the different external entities. The exception processor is consolidated on one machine which allows the sharing of functions that are common for different FTS applications.

5. The integration into one system of the two STAAR applications provides operational advantages. Since two VAX's will replace 7 PDP/11, there will be reductions in hardware maintenance costs, operation support, and physical plant requirements.

The variances from the conceptual model are:

1. A unified funds transfer database has been implemented which contains all the data that used to be segregated in the CIMS databases and that is now shared by the message processor, the transaction processor and the exception processor. However, this database is independent of all the other Financial Institution systems, and no sharing of data occurs between them. For example, the Funds Transfer and Letter of Credit systems are both part of the Transaction Processor. Even though in the conceptual model they are shown as sharing a database, no sharing is occurring at the present time, and each of them has its independent database.
2. There is no match with the Data Control component of the conceptual model. The Funds Transfer and Demand Deposit

Account systems have been implemented without Database Management System (DBMS) technology. Even though it is argued that a DBMS could not provide the high performance and recoverability needed by the Funds Transfer System to manage the transaction queues, DBMS technology could still be used to manage the dynamic files, (e.g. account balances) which would provide further advantages, especially for its integration with the Demand Deposit Account system. The Investigation system is the only one among the ones evaluated here, that uses DBMS technology, but this database is not shared by any other system.

3. The DDA system still communicates with other FI systems via tape hands-off. For example, a daily update of balances is sent to the Funds Transfer, Cash Management, and Letter of Credit systems.

Thus, even if there have been some gains in terms of integration, the main obstacle preventing the integration in the current implementation, is that the different systems have independent databases, and no DBMS technology is used.

B. AUTONOMY

The Funds Transfer and Demand Deposit Account systems have been implemented independently of each other and of the other FI systems. Furthermore, the Funds Transfer system and Demand Deposit Account implementation have been divided into

independent projects to permit, ease of implementation, obtaining immediate benefits, and accountability to individual managers. This matches the culture of the bank.

Even though in the implementation of the funds transfer system there has been a certain degree of centralization, (e.g., one processor for all CIMs, centralization of external interfaces, etc.), the banking groups still have logical control. Each of the banking groups have their own group of operators to perform data entry, repair, and verification of messages. Thus, even though all of the banking groups share the FTS Exception Processor, and there is a single queue containing all the messages to be repaired, each message will be repaired and verified only by the appropriate operator. From the point of view of the operators, their functions are performed as if each banking group had its own independent Exception Processor. Moreover, the terminals and operators are located in the banking group offices, and do not belong to the Financial Institution systems group.

The funds transfer system is an example where the hardware, the operation of the data center, the software development and maintenance, and the data control have been centralized but the business divisions still retain the management and control of the business operations. Thus, accomplishing both physical centralization and logical decentralization.

Applications have been hosted in independent hardware which follows the principle of autonomy. However, by "clustering" the applications in a VAX cluster, some integration is achieved, as it is the case with the Funds Transfer System.

Finally, even if the conceptual model calls for a shared database, the FTS and DDA have independent databases. This particular case illustrates that the bank's autonomy principle which emphasizes small projects outweighed the need for integration.

C. EVOLUTION

The new Funds Transfer system is better prepared to accomodate evolution. The single Transaction Processor and unified database provide flexibility if organizational changes need to be made in the bank which have an effect on the way customers are distributed by banking groups. The communication facility for inter-computer communication, DECNET/ETHERNET, has much more unused capacity allowing substantial communication growth with current technology.

The implementation of the Funds Transfer System in a DEC VAX cluster enables the capacity of the system to be increased by adding new hardware (e.g. processor, disks) to the VAX architecture. This can be accomplished without segregating the database since a single database can be shared by all the processors within the VAX cluster. For example, if the

processing capacity of the transaction processor (CIMS) is insufficient, a new VAX processor could be added to the cluster which would share the database. In the case of the Demand Deposit Account systems, capacity can be augmented by adding new hardware to the VAX architecture which would have been impossible with the previous PDP/11s. Moreover, programs can also be expanded for added functionality since there are no limitations regarding the maximum size of a program.

Chapter 7

Analyzing the Financial Institution Systems using the CIS model

In this chapter, an analysis of the Financial Institution systems is made using the CIS methodology. The CIS model for Financial Institution systems is illustrated in Figure 15. This model proposes a top down methodology where the strategic goals are first identified, then the composite information systems are analyzed focusing on technical and organizational obstacles that constrain the goal of integration. These obstacles are further evaluated in the next chapter. Finally, in chapter 9 alternative recommendations are made to attain the integration goal.

7.1 STRATEGIC GOALS OF FINANCIAL INSTITUTIONS

COST LEADERSHIP

According to Michael Porter [11], one way to gain competitive advantage is by being the cost leader in the industry. The demand for Financial Institution services has been increasing in the last years, thus the objective is to capture market share in the continuously growing market of Financial services. In order to achieve this, the Financial Institution Systems must be able to handle more volume without a corresponding cost increase. Several indications to

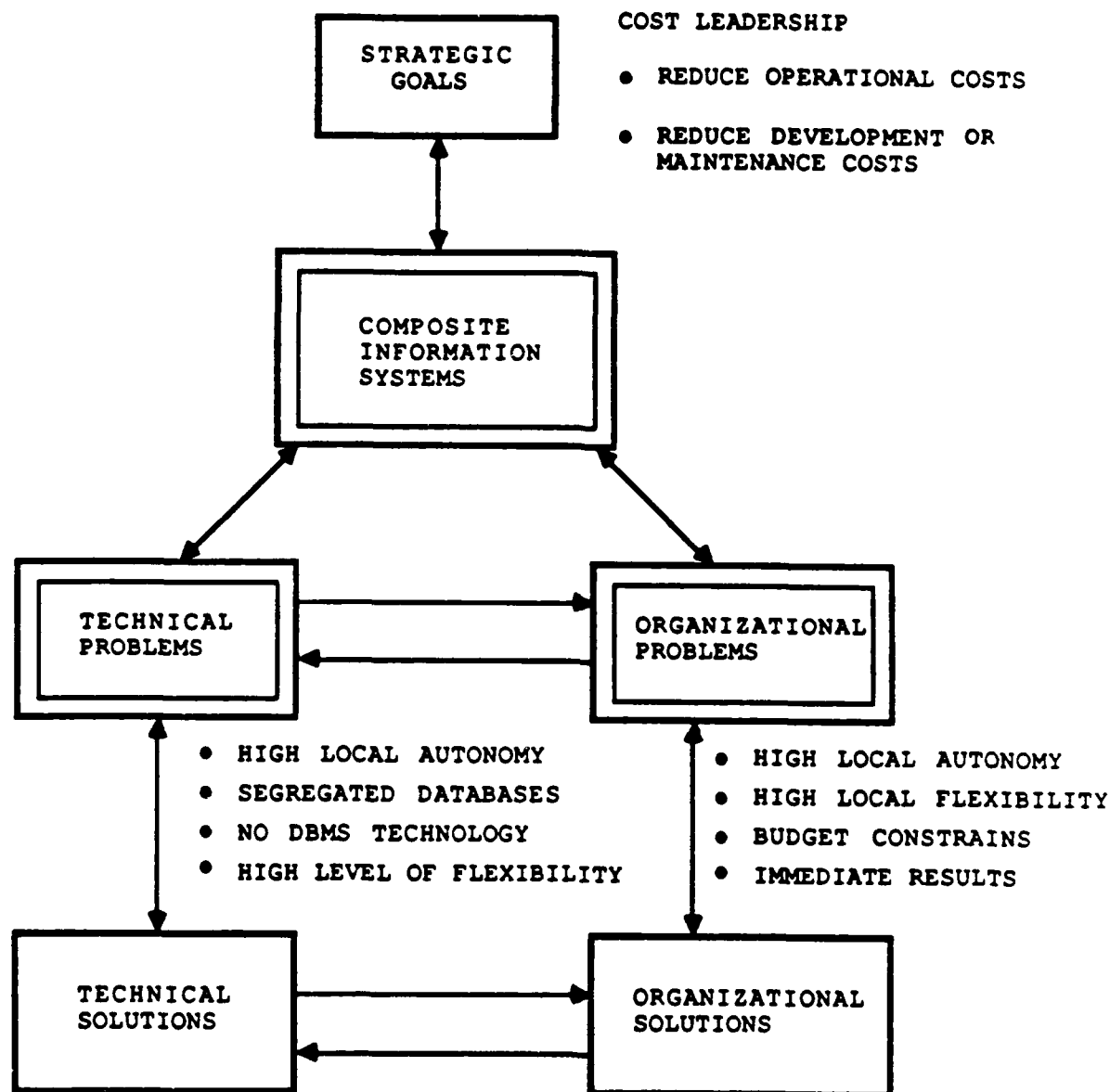


Figure 15

CIS Model for Financial Institution Systems

illustrating the increase in the demand of Financial services are: The number of funds transfer operations through CHIPS has been increasing at a rate of about 25% per year; there has also been an increased traffic of SWIFT messages for inter-bank customer balance and transaction reporting; and the volume of FED funds transfers which require immediate confirmation has increased considerably [9]. The systems objective is to accomodate this growth without incurring high operational, personnel, or maintenance costs. These costs could be controlled in the following areas:

1. Operational costs.

This category refers to the cost involved in running the data centers, the data entry, the verification functions, as well as tape hands-off. If these functions require a lot of manual intervention, the costs involved handling the increasing demand would increase proportionally with demand. For example, if confirmation of transactions were done manually because the dispersity of data prevents the possibility of real-time checks, the number of people required to handle the confirmations would increase as the demand grows.

An example where the bank has already taken action to reduce operational costs is found in the Funds Transfer System. An expert system was developed to support the exception processor functions. About 25% of the messages processed by FTS need some kind of manual intervention in order to be interpreted

and formatted. This 25% represents an average of 16,250 messages per day. The expert system helps to reduce the number of people needed to perform this task by doing the interpretation automatically. The task of the operator is then limited to checking the output produced by the expert system. This reduces dramatically the time required per unstructured message. Even if a high price was paid to develop the expert system, the savings in operational costs are well worth the additional initial expense.

Another example where costs can be reduced is by the elimination of tape hands-off to communicate across systems by merging redundant databases, or by using the communication network to communicate between systems. Not only do tape hands-off involve manual intervention, but also setting up procedures to handle it, and the possibility of introducing errors.

2. Development and/or maintenance costs.

There is a trade-off between initial development costs and maintenance costs. If systems have been developed to accomodate growth, and change, the up-front cost of development will most likely be higher whereas reducing the future cost of maintenance. Several examples are provided to illustrate this point. If the databases are dispersed, changes need to be propagated as they occur. Thus, procedures need to be set up and carried out to either make the change real time

or off-line as needed. If database technology is not used, a change in the data implies changing all the applications that use those databases. Therefore, changes to the software are needed. Moreover, whereas the price of hardware has been going down in the last years, the price of developing and maintaining software has shown an increasing trend.

7.2 CIS - THE FINANCIAL INSTITUTION SYSTEMS

As it was described in chapter 6, the Financial Institution Systems are still very independent of each other. So, although a common communication structure has been developed, most of them still have their independent databases which contain redundant data, thus constraining the goal of integration.

7.3 TECHNICAL OBSTACLES

The main technical obstacles towards integration that are found in the current implementation of Financial Institution systems are:

1. High level of autonomy.

Even though a strategic systems plan was developed in 1982 which resulted in the conceptual model discussed in chapter 5, most of the Financial Institution systems have still been developed independently. An example is provided by the Funds

Transfer system, and the Letter of Credit system. Although they both belong to the Transaction Processing component of the Conceptual model, they were developed independently resulting in two independent databases.

An example found in the current implementation where the principles of autonomy and integration are mediated is given by the implementation of the Funds Transfer system in a VAX cluster. The applications belonging to the Funds Transfer system have been hosted in independent hardware. However, these applications were grouped together in a VAX cluster which provided the ability to share data. The use of VAX clusters which has a maximum capacity of 16 nodes (i.e., processors, disk and communication controllers, etc.) allowed for independent hardware to be used for different applications. However, as more applications are added to this cluster, or as demand for FTS products grows, the capacity of the cluster may be exhausted, thus posing a technical problem since with the current implementation, data can only be shared within a cluster.

2. High level of flexibility.

The Financial Institution systems require a high level of flexibility. By keeping the systems small, higher flexibility is obtained but it presents problems for interconnecting these "small" systems. However, compared to the previous architecture, there has been a certain level of aggregation

providing gains in terms of integration capabilities.

3. Segregated databases.

The Financial Institution systems present the inefficient characteristic of having disperse and redundant databases. Each system has an independent database, even though some of the data is common across systems. This dispersity of databases represent the main obstacle towards integration. It also poses other problems, such as capacity and performance of the network and processors, duplicate processing and maintenance, etc. These problems are discussed in the next chapter.

4. Lack of use of Data Base Management Systems technology.

The lacking of DBMS technology in the current implementation has consequences on the maintenance of the databases. If no DBMS is used, and databases are shared among applications, update propagation, security, and consistency of the data needs to be controlled by the application. Moreover, changes to the database implies changing all the applications that access it. Note however that, for performance reasons, the use of databases is not always feasible in the bank. This represents a technical obstacle that can only be resolved with future advances in DBMS technology.

7.4 ORGANIZATIONAL OBSTACLES

The main organizational obstacles towards integration found in the current implementation of FI systems are:

1. High level of local autonomy.

The organizational structure of the bank is highly decentralized. Individual managers are given complete autonomy to make decisions, and operate their units. Consequently, managers are also fully accountable and responsible for meeting goals. Thus, they are reluctant to relinquish the control over their operations to other organizations in the bank. An example to illustrate this point is provided by the development of the historical database (HDB), and the MIS system. These two applications, which started at the same time, ended up going in different directions [8]. Due to the lack of communication and trust among these two groups, the MIS system ended up obtaining this information from many different applications. In the process having to perform several layers of aggregation. Organizational obstacles prevented the MIS application from taking some of the aggregated data from a single source, the historical database. This in turn led to duplication and inefficient resource allocation.

2. High level of local flexibility.

Given the decentralized philosophy of the bank, individual

managers want to have the maximum amount of flexibility to accomodate change and growth.

3. Immediate results.

The culture of the bank is to engage in projects that provide immediate benefits. This has a disadvantage for implementing any long term plan that would provide benefits in the long run, such as planning for ease and reduction of future maintenance costs versus initial development costs.

4. Budget constraints.

Budget constraints jeopardize the goal of integration since given the dispersity of databases, their integration may require a high up-front cost. The goal of the business divisions is to get the work done. This leads to inefficiencies in the implementation and operation of systems. An example where considering only immediate results, and operating under budget constraints had a negative effect, is in the lack of integration of the data. Sharing the data that is used by several applications is a very desirable goal. This however, would have involved an initial high cost which under these constraints was impossible.

After reviewing the CIS concept and applying it to the Financial Institution Systems, the following chapters will analyze and discuss the technical obstacles in the current implementation, and some alternatives for future directions.

Chapter 8

Evaluation of the current implementation of Financial Institution Systems

After having analyzed the systems composing the Financial Institution Systems in chapter 6, this chapter presents an analysis of the integration capabilities across the FI systems using the CIS methodology.

8.1 INTEGRATION CHARACTERISTICS OF FI SYSTEMS

The current implementation of the Financial Institution Systems present the following characteristics:

1. Common communication facility. The centralized network facility (DECNET/ETHERNET) enables the exchange of messages across Financial Institution's systems, i.e., Funds Transfer, Letter of credit, Cash Management, Investigations, and Demand Deposit Account. For example, a funds transfer operation which generates funds or receives incoming funds for a bank's customer, produces one or more transactions that are automatically routed to the proper application to either credit an incoming fund or debit the account for a funds transfer.
2. Segregated databases. All the systems, Funds Transfer, Cash

Management, Demand Deposit, and Investigations have their independent databases as illustrated in figure 16. All these databases contain redundant information, i.e. some databases are partially a "shadow database" of the original database. The redundant information in the "shadow databases" includes, static customer information such as, customer name, account number, etc, and dynamic information such as, customer account balances, authorizations, etc.

8.2 PROBLEMS POSED BY CURRENT IMPLEMENTATION OF FI SYSTEMS.

The main problem posed by the current implementation is the dispersion and redundancy of data which generates "shadow databases." A discussion of the implications that these "shadow databases" have on performance, consistency of data, systems development, and systems maintenance is presented in the next section. All of these have an effect on the cost of operation, and maintenance of the Financial Institution Systems.

In the Financial Institution Systems, updates are classified as either real-time updates or off-line updates. In some cases, "shadow databases" require real-time updates which implies that a change in one database needs to be propagated to the "shadow databases" as soon as it occurs. In other cases, changes to a database do not require to be propagated in real time, but they can be done off-line at a later time.

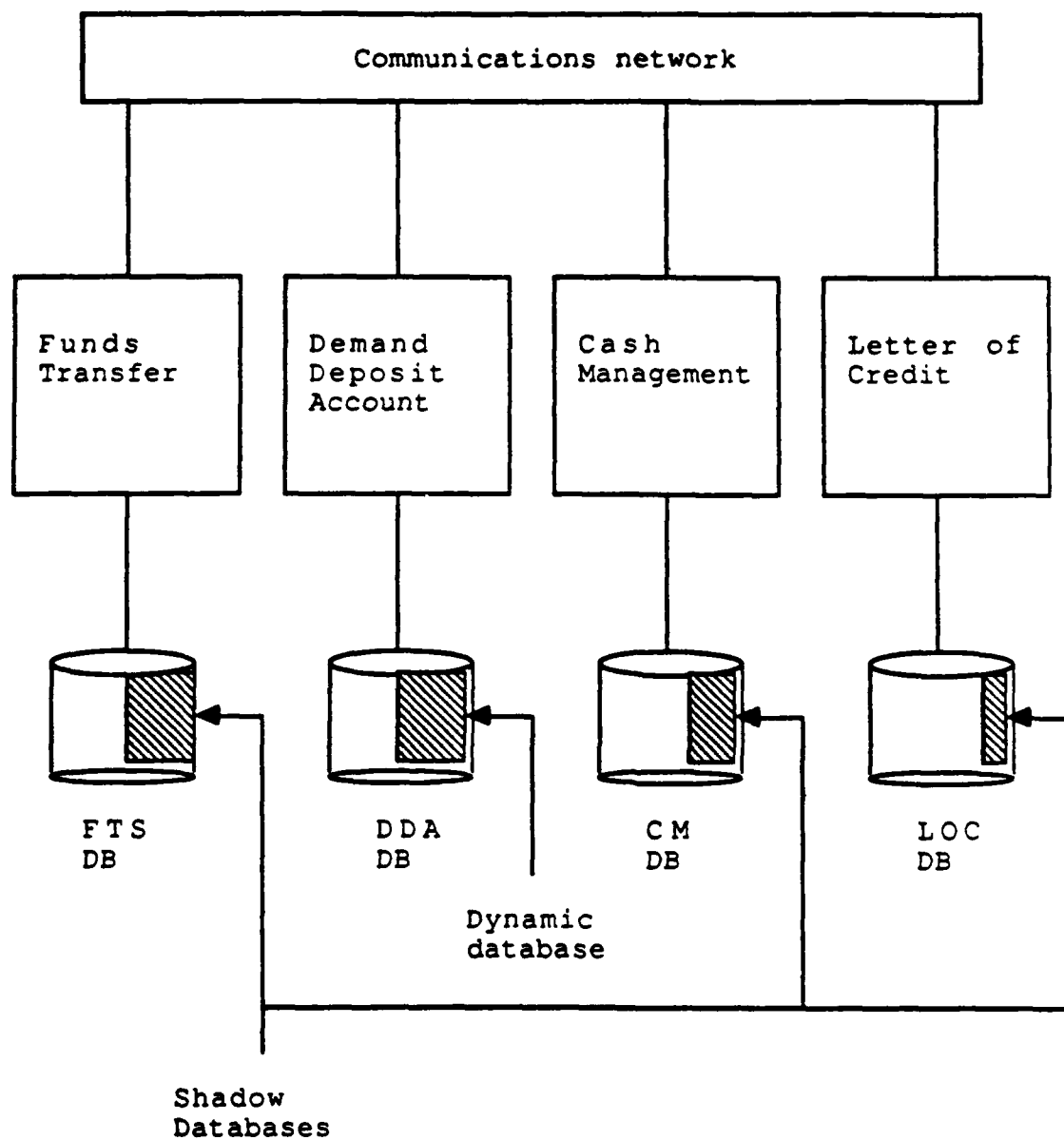


Figure 16

Applications with independent databases

In the latter case, procedures need to be set up and carried out on periodical basis. However, either case requires some kind of communication among the Financial Institution Systems to propagate the updates. It can vary from sending a message through the network in real time to tape hands-off.

8.2.1 REAL-TIME UPDATES

An example of a real time update is given by the funds transfer operation which spawns messages to other Financial Institution systems with the sole purpose of updating "shadow databases". There are two types of funds transfer transactions: 1) Transactions that change the bank books since one of the players in the funds transfer is a customer of the bank. And 2) Those transactions that do not change the bank books since the bank is acting as an intermediary to other bank's customers.

Currently, each funds transfer transaction which generates funds or receives incoming funds for a bank's customer, (i.e., requires a change on the bank books), spawns at least two messages to other Financial Institution systems. These messages are called "shadow postings". The "shadow postings" generated by a Funds Transfer transaction is illustrated in Figure 17. The messages spawned by FTS are:

- Two (2) posting messages to the DDA system.
- One (1) posting message to the Cash Management system, in

Funds transfer
transaction
(DDA and CM
customer)

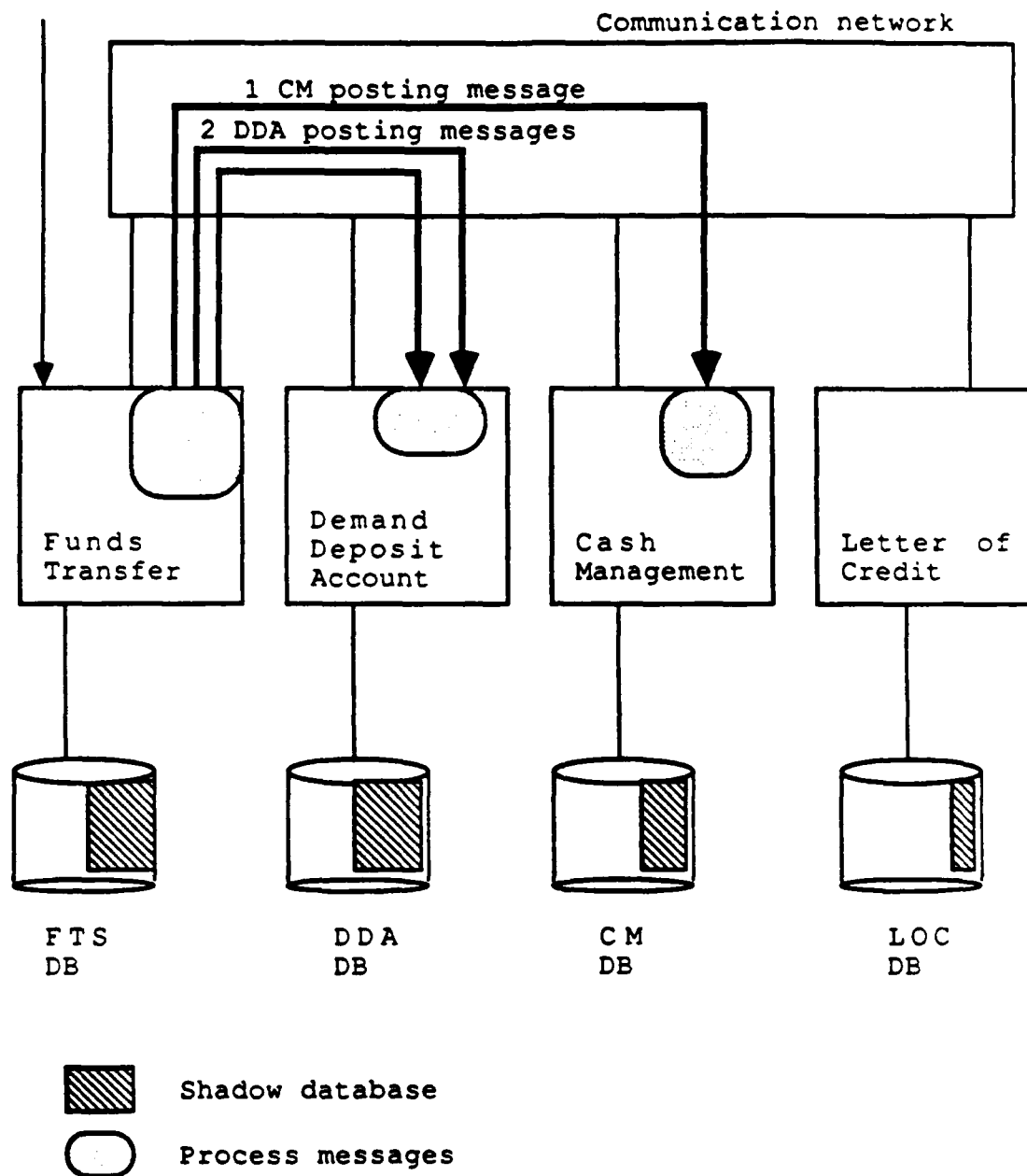


Figure 17

Funds transfer transaction generating
posting messages

the case that the customer is also a Cash Management customer.

The sending of messages is necessary since each application keeps in its independent database a record of the customer status (e.g., the account balance,) and any change to the customer status needs to be propagated in real time to all "shadow databases" (e.g., Demand Deposit Account and Cash Management databases). However, the bank also acts as an intermediary for other banks; in those cases, it is not necessary for the funds transfer transactions to send a message to either the DDA or CM systems.

Assuming that 50% of funds transfers are for other banks, in average, for every funds transfer transaction processed, $1 \frac{1}{5}$ transactions are spawned to other FI systems. The average $1 \frac{1}{5}$ transactions spawned by the Funds Transfer system can be broken down into two components: 1) One (1) posting message transaction to the Demand Deposit Account System; 2) $1 \frac{1}{5}$ posting message transaction to the Cash Management System (i.e., in average, 20% of the funds transfer transactions are for Cash Management customers).

Thus, given that the average number of funds transfer transactions per day is currently 60,000 transactions, the average number of messages (i.e., shadow postings) spawned by the Funds Transfer system in real time to update "shadow

databases" is:

- 60,000 messages to the Demand Deposit Account System;
- 12,000 messages to the Cash Management System.

8.2.2 OFF-LINE UPDATES

Other "shadow databases" updates are not required in real time. Some examples in this category are, the maintenance of static customer information such as, addition of a new customer, change in customer address, and the reconciliation of customer balances. The latter is done daily by DDA, and produces a tape that is sent daily to all FTS, CM, and LOC. Thus, procedures need to be set up to propagate the updates as frequently as necessary. This usually requires manual intervention, e.g. tape hands-off.

8.3 IMPLICATIONS OF THE CURRENT IMPLEMENTATION OF FI SYSTEMS

The current implementation has implications on the performance, maintenance, and operation of the FI systems. These are summarized in figure 18.

1. Network Performance.

In average, there are 72,000 messages flowing across the network with the only purpose of propagating updates to

Issues	Current implementation
Message posting	60,000 from FTS to DDA 12,000 from FTS to CM
Processing overhead	Considerable
DDA performance	Good
CM performance	Good
Tape hands-off	1 tape from DDA to: FTS CM LOC
Off-line updates	Send transaction file from FTS to Investigations
Duplicate maintenance	Maintenance of Static DB in: FTS, DDA, CM, LOC, and Investigations systems
Duplicate Information	Dynamic and Static DB in: FTS, DDA, CM, and LOC

Figure 18

Current implementation issues

"shadow databases". The flow of messages across systems is proportional to the number of funds transfer transactions. Thus, as the number of funds transfer transactions is growing at a rate of 25% per year, the flow of messages will increase at the same rate. This increasing number of messages flowing through the network, may have a negative effect on the capacity and performance of the network.

2. Processing overhead.

In order to propagate updates to "shadow databases" in real time, the applications need to have software (i.e. application's code) to make decisions on whether or not to send a message to other applications, and to receive the messages sent by other applications. For example, for any funds transfer transaction processed, decisions are made in the FTS application program on whether to send a message to either DDA, or to CM, depending on the customer status. If the bank books need to be changed as the result of a funds transfer operation, the appropriate posting messages need to be constructed and be sent to the appropriate application(s). All these require the use of processing time to perform functions which are not relevant to the funds transfer operation. It also requires the development and maintenance of the additional software to handle the sending and receiving of messages.

Taking into account that the average number of funds transfer

transactions is about 60,000, this could mean a lot of processing time with the solely purpose of propagating updates. On the other hand, the DDA and CM applications need to have code to receive and process the update messages that are sent by FTS.

3. Tape hands-off.

Some update information is transmitted via batch. At night, posting and balance updates are sent via tape from the Demand Deposit Account system to the Funds Transfer system, the Cash Management system, and the Letter of Credit system. To perform this updates via tape hands-off implies that procedures need to be set-up (i.e., development of software), and carried out which involves the intervention of an operator.

4. Off-line update propagation.

Every night, an ASCII file with the transaction history of the day is sent via the communications network from the Funds Transfer System to the Transaction Investigation system to update the Historical Database. However, this seems like a plausible approach since the Funds Transfer database contains only the transactions that are currently being processed (i.e. today's transactions) whereas the historical database contains a transaction history for the 90 previous days. Updating the historical database real-time would generate a tremendous amount of messages that would not provide greater benefits.

On the other hand, the historical database is for obtaining information rather than processing transactions, so this application belongs to the Information Processing component of the conceptual model described in chapter 5. Sharing the historical database with the Transaction Processing database is not possible because first, these databases serve different purposes, and second, the VAX cluster has a maximum of 16 nodes and not all of the Financial Institution systems can fit in the same cluster.

As the transfer of daily transactions from the FTS database to the historical database is done daily at night, the historical database contains information up to the previous day. Thus, if there is any investigation that requires looking at today's data, the AIRS operator will have to query the Funds Transfer database.

5. Duplicate maintenance.

The databases contain other redundant information which is not required to be updated in real-time. However, procedures need to be set up and carried out to propagate any change made to the original database duplicating the database maintenance. For example, if there is a change in static information such as, a customer address is changed, or a new customer is added to the database, a procedure is needed to propagate this change to "shadow databases". In these cases, it is usually a manual intervention to either start a procedure or to perform

the change manually across the databases. Currently in the bank, all the changes to static information are performed manually by an operator. Moreover, since the change needs to be transmitted to the disperse databases, this information is manually entered into all the systems that use this information, (i.e., Funds Transfer, Demand Deposit Account, Transaction Investigation, and Cash Management). Operators belonging to different divisions will perform the same function to enter the new information in the different databases. Furthermore, this operation is prone to errors.

6. Duplicate information.

The Funds Transfer system besides having redundant information about the customer, name, address, account balances, etc, needs to have information about which customers are DDA and which are CM customers. This implies further use and waste of storage.

7. Integration of FTS and Letter of Credit.

The transaction processor in the conceptual model is composed of two systems: the Funds Transfer and the Letter of Credit system. In the conceptual model, these systems share the same database but in the current implementation, they both have independent databases. Thus, information about funds transfers is not known to the letter of credit system until the next day, and viceversa.

All these factors, processing overhead, tape hands-off, duplicate maintenance, etc. have an effect on the cost of operation and maintenance of the FI systems.

8.4 COSTS OF OPERATION AND MAINTENANCE OF THE REDUNDANT DATABASES.

The total costs of operation and maintenance of the redundant databases, namely the dynamic database, and the static database can be quantified in terms of, processing costs, personnel costs, storage, etc. and they are shown in the cost function below. The cost of operating a system is shown as:

$$C_o = C_{om} + C_{others} \quad (1)$$

The cost of operation and maintenance is given by the following equation:

$$\begin{aligned} C_{om} &= F(P, T, OL, DM, DI) \\ &= P + T + OL + DM + DI \end{aligned} \quad (2)$$

where

P = processing overhead,

T = tape hands-off,

OL = Off-line updates through network,

DM = Duplicate maintenance, and

DI = Duplicate information.

This cost function is additive, since in the current implementation, all these costs are present. Moreover, the dynamic database is partially redundant in the Funds Transfer system, the Demand Deposit Account System, the Cash Management system, and the Letter of Credit system.

The definition of P, T, OL, DM, and DI follows,

1. Processing overhead (P). Processing overhead is the amount of processing spent on exchanging messages to propagate updates in real time. Processing overhead is a concern since it constraints the ability to accomodate growth without having to migrate to a more powerful processor. Also, removing processing overhead provides the capability to have processing slack that can be used to host other applications, thus using resources more efficiently.

$$P = P_f + P_d + P_c \quad (3)$$

$$P_f = [(N_f * t_{f1}) + (N_f * (N_d + N_c) * t_{f2})] * k_p \quad (4)$$

$$P_d = (N_f * N_d * t_d) * k_p \quad (5)$$

$$P_c = (N_f * N_c * t_c) * k_p \quad (6)$$

where

P_f = Processing overhead in Funds Transfer System.

P_d = Processing overhead in Demand Deposit Account System.

P_c = Processing overhead in Cash Management System.

N_f = Total number of funds transfer transactions which are processed daily. This includes those transactions that spawn messages to other applications, and those transactions that do not spawn any message.

N_d = Percentage of funds transfer transactions which spawn a message to DDA. The current average is 100%, thus N_d would be equal to 1.

N_c = Percentage of funds transfer transactions which spawn a message to CM. The current average is 20%, thus N_c would be equal to 0.20.

t_{f1} = Units of time required by FTS to decide whether a particular transaction requires a message to be spawned.

t_{f2} = Units of time required for FTS to build and send a message to another application.

t_d = Units of time required by DDA to process the receiving message and update its database.

t_c = Units of time required by CM to process the receiving message and update its database.

k_p = Average cost (in dollars or fraction) per unit of processing time.

2. Tape hands-off (T). Tape hands-off refers to the communication via tapes among FI systems to propagate updates which are not required in real time.

$$T = A * n_t \quad (7)$$

$$A = (n_A * t_A * k_A) + (t_t * k_p)$$

where

n_t = Number of tapes to be exchanged among FI systems.

n_A = Number of operators needed to handle a tape. For example, if a tape is sent to three different systems located in different sites, four operators would be needed.

t_A = Units of time required by an operator to handle one tape.

t_t = Units of processing time required to process one tape.

k_A = Average cost (in dollars or fraction) per unit of operator's time.

k_p = Average cost (in dollars or fraction) per unit of processing time.

3. Off-line updates (OL). Off-line updates refers to the exchanging of files across FI systems (through the network) to propagate updates which are not done in real time.

$$OL = \sum_{i=1}^N B_i * [(t_1 + t_2) * k_p] \quad (8)$$

where

N = Number of files to be transferred across FI systems.

B_i = Number of blocks of file i .

- t_1 = Units of processing time required by the originating application to create and transfer one block.
 t_2 = Units of processing time required by the destination application to receive and process one block.
 k_p = Average cost (in dollars or fraction) per unit of processor time.

4. Duplicate maintenance (DM). Duplicate maintenance refers to the manual maintenance of redundant databases. One example is provided by the maintenance of static information which is manually done by operators in each one of the databases.

$$DM = M * n_b * n_c * (1 + p) \quad (9)$$

$$M = (t_M * k_M) + (t_c * k_p)$$

where

- n_c = Number of changes to be performed to one database/day.
 t_M = Units of time required by one operator to make one change.
 t_c = Units of processing time required to process the change.
 k_M = Average cost (in dollars or fraction) per unit of operator's time.
 k_p = Average cost (in dollars or fraction) per unit of processing time.
 n_b = Number of databases to be updated. It is assumed that one operator is required per database. Note that as databases may belong to different divisions, different

operators perform the change in each of the databases.

p = The probability of making at least one error while performing a change in any of the databases. This probability has been calculated to be p (1).

4. Duplicate information (DI). Duplicate information refers to the cost incurred on storage in order to maintain redundant databases.

$$DI = (n_b - 1) * B * k_s \quad (7)$$

where

n_b = Number of databases that contain redundant information.

B = Number of blocks.

k_s = Average cost (in dollars or fraction) per block of storage.

This model can be used to compute the cost of maintaining and operating the current implementation which contains redundant databases.

(1) Assume that we have n operators. The probability of one operator making an error is p , being p very small. Then, the probability of having one operator making no errors will be $(1-p)$. Thus, the probability of having no errors by any operator will be $(1-p)^n$. Therefore, the probability of having at least one error will be $1 - (1-p)^n$, which is approximately np when p is very small, as can be seen by using the binomial expansion.

In sum, the current architecture presents inherent drawbacks given that data is disperse and belongs to each application. Manual procedures and tape hands-off are still needed to accomplish propagation of updates. For real time updates, a continuous flow of messages is needed to transmit messages across applications. As can be seen in equation (3) above, as the number of funds transfer transactions increase, the processing overhead would increase linearly. All these represent costs that could be diminish with other approaches to be explored in the next chapter.

Chapter 9

Future Directions

As it was discussed in the previous chapter, the main drawback of current Financial Institution Systems is their disperse and redundant databases which require either manual procedures or a continuous flow of messages to propagate updates.

An alternative to the current implementation is to have a single database containing all the information that is otherwise redundant across the databases, or to have distributed databases that can be accessed by systems located in different hardwares. The consolidated database would be accessed by all the systems that need the information contained within it. Two approaches are then possible, one is to consolidate the data without sharing it, so that requests and updates will be done by exchanging messages across systems. The second approach is to share the databases that contain redundant information. The different alternatives considered are explained below.

9.1 CONSOLIDATION OF DYNAMIC DATABASE

This approach calls for the consolidation of information that is redundant across databases. In particular, the database

containing dynamic information (e.g., account balances, authorizations, etc.) would be consolidated in one of the FI systems. One of the following two systems can be chosen as the host to consolidate this database: 1) the Demand Deposit Account System, or 2) the Funds Transfer System. The host system is the only one owning the database. The other FI systems would not share the consolidated database, but instead they would access it by generating a message directed to the host system to either request information, or update the database.

In this case, the Demand Deposit Account system has been chosen as the host for dynamic information. The main reason to consolidate the dynamic database in this manner is the very nature of this data. The DDA system produces periodic reports which are mainly batch on balance information, a lot of them being produced daily. Having this information in the Funds Transfer System would imply a continuous stream of messages that would degrade the DDA processing and would delay its delivery schedules. As it was mentioned in chapter 6, meeting deliverables on time are critical for DDA. Figure 19 provides a summary which compares the merits of the consolidation of Dynamic data in either DDA or FTS to the current implementation.

Issues	Current implementation	Consolidated DB in DDA	Consolidated DB in FTS
Message exchange and shadow posting	From FTS: 60,000 to DDA 12,000 to CM	>60,000 from FTS to DDA 6,000 from CM to DDA	None from FTS or DDA 6,000 from CM to DDA
Processing overhead	Considerable	Considerable	Less
DDA performance	Good	Good	Bad
CM performance	Good	Less good	Less good
Tape hands-off	1 tape from DDA to: FTS CM LOC	None	None
Off-line updates	Send transaction file from FTS to Investigations	Same	Same
Duplicate maintenance	Static DB in: FTS, DDA, CM, LOC, and Investigations systems	Almost same	Almost same
Duplicate Information	Dynamic and Static DB in: FTS, DDA, CM, and LOC	None	None

Figure 19

Comparison of consolidation of dynamic database
vs. current implementation

9.1.1 PROBLEMS SOLVED BY CONSOLIDATING THE DYNAMIC DATABASE

1. Eliminating tape hands-off among applications.

The daily posting and balance update that is sent via tape from the Demand Deposit Account system to the Cash Management system, Funds Transfer system, and Letter of Credit system is eliminated since the database would be consolidated in DDA. Thus, tape hands-off that involve manual intervention and setting of procedures are further eliminated from the FI systems.

2. Some Reduction in the propagation of updates.

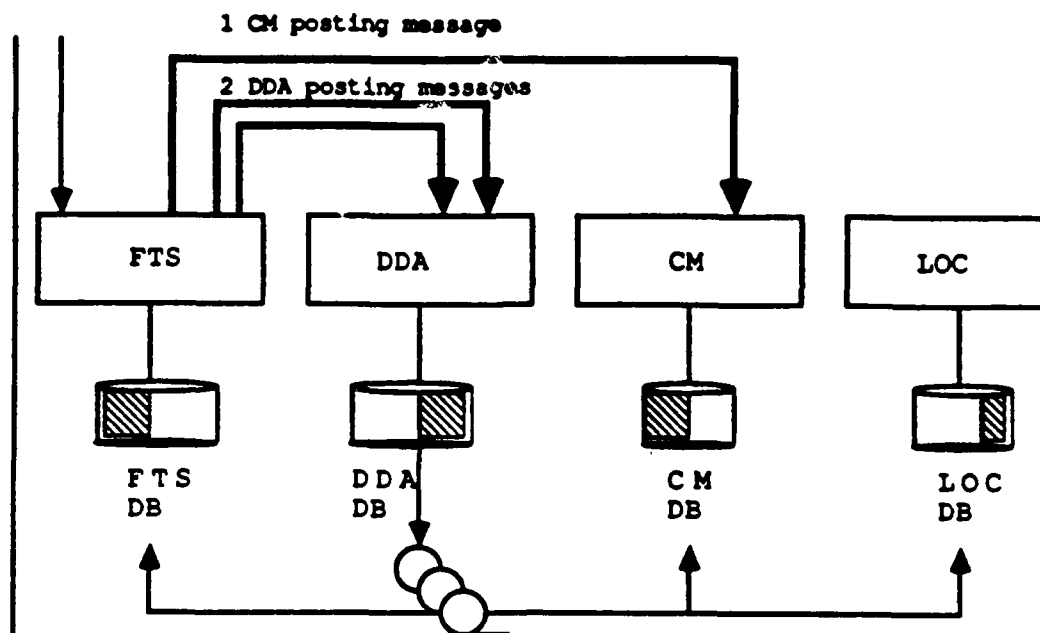
At the moment, there is a flow of messages throughout the network with the sole purpose of propagating updates. In this alternative approach, inquiry messages will be generated on demand. This is to say, messages would be generated to either request information, whenever needed, or to update the bank books in real-time. A comparison of the flow of messages in the current implementation and in the consolidation of dynamic data in DDA is shown in Figure 20.

A. Exchange of Messages between FTS and Cash Management.

Currently, 20% of the transactions processed by the Funds Transfer System require a message to be sent to the Cash Management System to update its database. This represents an average of 12,000 messages that are sent daily. However, Cash Management only performs inquiries, not updates, and

Funds transfer
transaction
(DDA and CM
customer)

CURRENT IMPLEMENTATION



CONSOLIDATED DATABASE

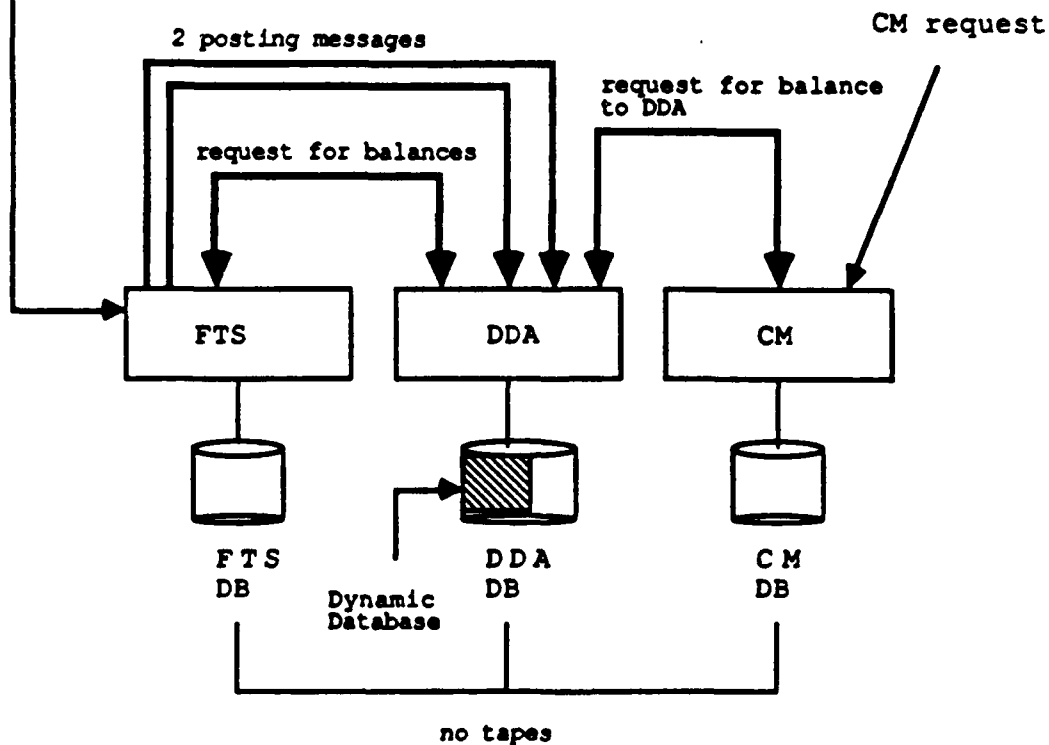


Figure 20

Comparison of message exchanging

the average number of Cash Management inquiries is 6,000. With the new approach, the flow of messages from the Funds Transfer system to the Cash Management System to update the CM database will be eliminated, and only the demanded number of messages from Cash Management (i.e. an average of 6,000) will be generated. Therefore, the flowing of messages between FTS and CM is reduced by 50%.

B. Exchange of messages between FTS and DDA. The funds transfer system currently spawns an average of 60,000 messages to DDA. This number of messages would still be necessary since the dynamic database is consolidated in DDA. Moreover, more messages are needed to request information that was previously available in the redundant FTS database. However, a gain is still achieved since posting to the bank books would occur in real time, that means funds transfer and DDA transactions will be checking balances and overdraft authorizations against the real customer balances, (i.e., the bank books). Currently, funds transfer transactions checks are done against the previous day balances.

The two main gains obtained from this approach are, to reduce tape hands-off and duplicate storage, and performing overdraft and balance checking against real-time information.

9.1.2 PROBLEMS NOT SOLVED BY THIS APPROACH

On the other hand, the following problems would not be solved by this approach:

1. Maintenance of static information.

Static information, customer names, addresses, etc. would still have to be maintained in the Funds Transfer System. The main reason is for performance. Currently, the Funds Transfer system maintains these tables in real memory for quick access. This information is used by the Message Processor, the Transaction Processor, and the Exception Processor. If the information is consolidated in DDA, FTS would have to send messages to DDA to obtain it, degrading the FTS performance. On the other hand, consolidating this information in FTS would degrade the reporting system that DDA handles.

However, as the Cash Management system is an on-line systems which mainly performs queries on balances, and it will be accessing the consolidated database, it does not need to have the static information locally, so the problem would be partially solved by eliminating one "shadow database", (i.e., the CM static database).

2. Communication with the Transaction Investigation System.

The daily transmission of the transaction history (i.e., to update the historical database) which is sent from the Funds

Transfer System to the Transaction Investigation System would also not be eliminated.

3. Processing overhead.

The processing time and code required to send update messages from the Funds Transfer System to the Cash Management System would be eliminated. However, new code would have to be written to coordinate queries from Cash Management to be handled by DDA. The amount of code and processing time to manage the exchange of messages between FTS and DDA will be the same.

Even though the suggested approach does not represent the ideal system, it could be implemented as a transitory stage for the next proposed alternative which calls for a more cohesive integration (e.g., shared data resource) of the Financial Institution Systems.

9.1.3 COST OF MAINTENANCE AND OPERATION.

The cost function shown in chapter 8 will be affected as follows,

$$C_{om} = F (P , T , OL , DM , DI)$$

$$= P + T + OL + DM + DI$$

T will be reduced in more than 50% since tape hands-off are dramatically reduced in order to propagate updates.

DI will be reduced in almost 100% since duplicate storage will be almost eliminated.

Therefore, the costs of operation and maintenance are reduced, and other intangible gains such as, checking against the current bank books are also obtained, although they are not quantified in the cost function above.

9.2 SHARING THE DYNAMIC DATABASE.

This approach calls for the consolidation and sharing of dynamic information. There are two locations where this data could be kept: 1) as part of the Transaction Processing component; 2) as part of the Information Processing component. The approach to be analyzed now is the first one, to locate the consolidated daily dynamic data as part of the Transaction Processor. The other approach is analyzed under the new technologies section.

With the current architecture, applications need to be in the same cluster in order to be able to share a database, thus by placing applications in the same cluster, sharing of data can

be accomplished. However, only the Funds Transfer system and the Letter of Credit system are contained within the Transaction Processing component, and the Letter of Credit (LOC) system is not part of the cluster. So, DDA, LOC, and CM would have to be moved into the Transaction Processing component. The movement of the same-day processing DDA to TP was included in the long-term Strategic Systems Plan for Financial Institution systems. However, in this case, the Cash Management system would also have to be moved into TP in order to be able to share this database. If CM is not moved to TP, messages will have to be sent on request in order to obtain the desired information. This is a plausible alternative since at the moment there are approximately 6,000 of these requests.

On the other hand, if DDA, LOC, and CM form part of the TP cluster (currently named, Funds Transfer cluster), the dynamic database would be consolidated and shared by FTS, CM, LOC, and DDA. This approach provides several advantages: performing all balance and overdraft validation against current data; drastic reduction in the number of messages flowing through the network, and reduction on duplicate maintenance and tape hands-off. Figure 21 illustrates this architecture, and Figure 22 presents a summary of the merits of this approach compared to the current implementation. An explanation of the pros and cons of this approach follows.

9.2.1 PROBLEMS SOLVED BY THIS APPROACH

1. Drastic reduction in update propagation.

At the moment, there is a flow of messages throughout the network with the solely purpose of tracking changes for the segregated dynamic databases. In this new approach, no exchange of messages will be necessary for this purpose. Moreover, updates of the bank books and verification against the bank books would be done in real time.

A. Messages from FTS to Cash Management.

As explained before, the Funds Transfer System sends an average of 12,000 messages per day. This message flow could now be completely eliminated since the Cash Management system would be sharing the database that also contains the bank books (e.g., account balances).

B. Messages from FTS to DDA. The flow of messages from Funds Transfer to DDA, would be completely eliminated since the Demand Deposit Account and the Funds Transfer System will be sharing the same database and FTS will do the postings against this database. Moreover, the funds transfer transactions will be checking overdraft and balances against the actual customer balance.

2. Eliminating tape hands-off across applications.

The daily posting and balance update that is sent via tape

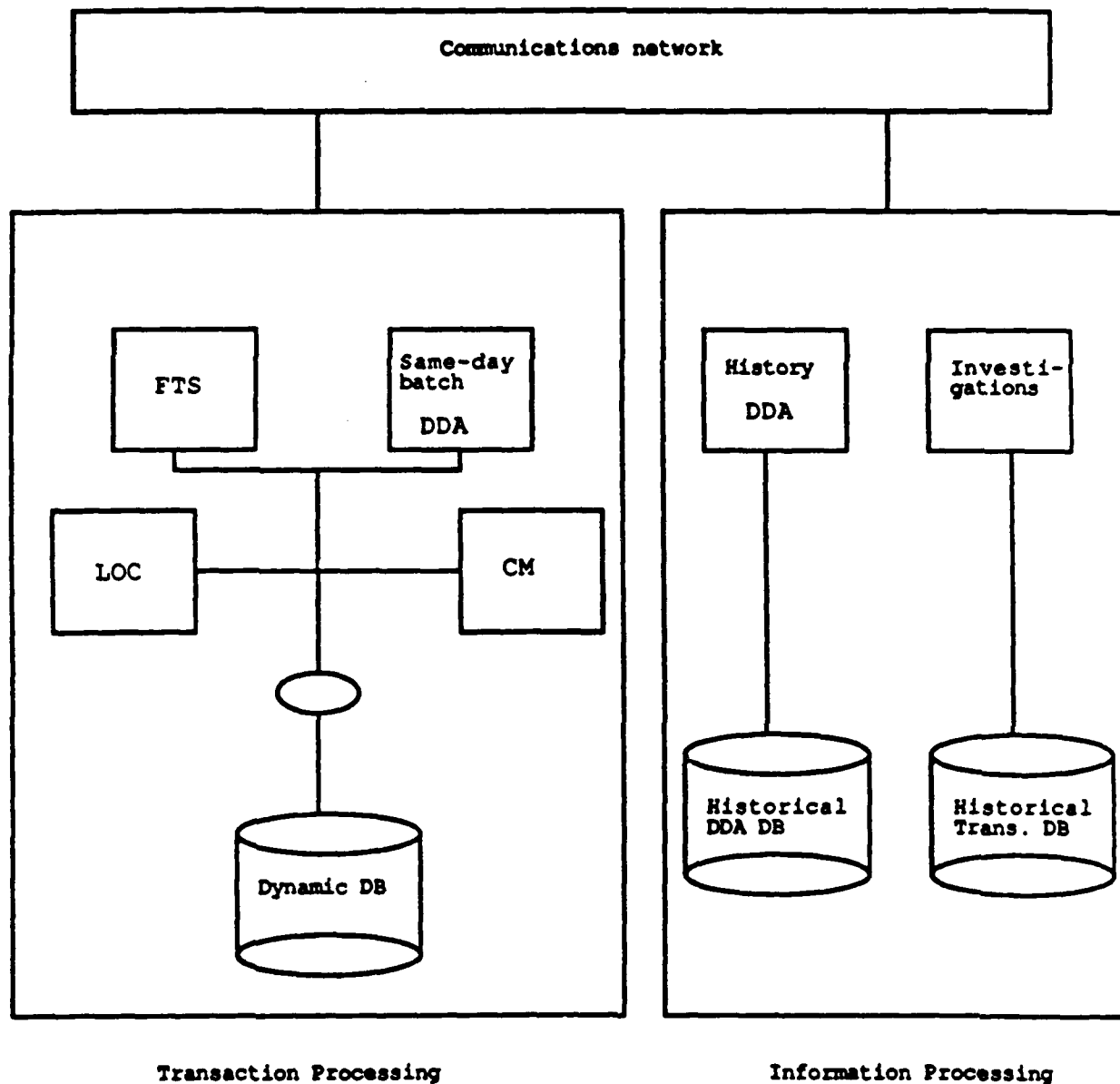


Figure 21

Sharing of daily dynamic database

Issues	Current implementation	Share dynamic DB	
Message exchange and shadow posting	From FTS: 60,000 to DDA 12,000 to CM	None	←
Processing overhead	Considerable	None	←
DDA performance	Good	Good	
CM performance	Good	Good	
Tape hands-off	Tapes from DDA to: FTS CM LOC	Tapes from TP to: Historic DDA	←
Off-line updates	Send transaction file from FTS to Investigations	Same	
Duplicate maintenance	Static DB in: FTS, DDA, CM, LOC, and Investigations systems	Reduced	←
Duplicate Information	Dynamic and Static DB in: FTS, DDA, CM, and LOC	None	←

Figure 22

Comparison of sharing dynamic database
vs. current implementation

from the Demand Deposit Account system to the Funds Transfer System, Cash Management, and Letter of Credit systems would be eliminated since the data is consolidated in the TP cluster. However, the daily posting and balance update will have to be sent to update the historic DDA database which resides in the Information Processing component. This would be unavoidable since the databases contain different kind of information.

3. Eliminating duplicate processing.

The processing time and code required to send update messages from the Funds Transfer System to the Demand Deposit Account and Cash Management System, and the code in DDA and CM to receive the messages would be completely eliminated.

4. Reduce duplicate maintenance.

The duplicate maintenance that takes place to update static information would be completely eliminated for all applications within the Transaction Processing. Thus, no duplicated maintenance is needed for CM, daily DDA, and FTS. However, now any change made to static information needs to be reflected in the DDA historical database, and in the Transaction Investigations historical database. This can not be eliminated because the nature of the data contained in these databases is different.

9.2.2 PROBLEMS NOT SOLVED BY THIS APPROACH

1. Cash Management System.

A new problem is created since the historic database will remain in the Information Processing component, and some of the Cash Management inquiries would require to get a transaction history. In this were the case, a message would have to be sent to IP to respond to the transaction history inquiry. The number of requests of this type is currently less than 6,000.

2. Duplicate maintenance for historical databases.

As the historical and dynamic databases are different in nature, some duplicate maintenance will still take place. For example, a daily transaction file needs to be sent from FTS to be the Investigations system to update the historical database. Second, the daily postings made to the dynamic database have to be sent to the DDA historic database. This could be accomplished by sending a file through the network instead of doing it via tape hands-off. Finally, maintenance to the static database has to be performed in both sites, the transaction processing and the information processing. Note that less static databases need to be maintained.

9.2.3 COSTS OF MAINTENANCE AND OPERATION

The cost function shown in chapter 8 will be affected as

follows,

$$\begin{aligned} C_{om} &= F(P, T, OL, DM, DI) \\ &= P + T + OL + DM + DI \end{aligned}$$

P will be reduced by 100% since the number of messages flowing through the system because of update propagation is completely eliminated.

T is reduced by at least 75% since the number of updates via tapes has also been reduced.

OL would be the same, and may increase if the daily DDA is sent through the network to the historic DDA.

DM is also reduced by about 50% due to a reduction in the maintenance of static information.

DI is reduced by 100% since storage is no longer needed to hold redundant databases.

Therefore, this approach shows to reduce the operation and maintenance costs significantly. The main advantages are, to enable growth with current architecture and technology, reduce personnel costs, and storage costs.

9.2.4 DBMS VS. NO DBMS APPROACH

If the dynamic database is implemented without a database management system, a few problems are encountered. First, since different applications are sharing the same database, namely the dynamic database, concurrency has to be done at the application level, i.e., code needs to be written into the applications to perform concurrency control. Second, applications need to know exactly where the data is, so there is no transparency in the access to data. Third, security access to the database has also to be controlled at the application level. Fourth, the benefits that DBMS provides for reporting and screen formatters that speed the development process are not utilized. For all the reasons stated above, and given that the performance required to access this database is well within the limits provided by current DBMS, it is desirable to implement this database using DBMS technology. The use of DBMS will provide advantages for maintenance, and speed of development since the programmer would be relieved of coding tasks that would be taken care of by the DBMS.

9.3 NEW TECHNOLOGIES - DISTRIBUTED DATABASES

A new technology to be explored here is distributed databases. Currently, two commercial packages are available that work in

the VAX environments, Distributed Ingress and Distributed Oracle. This technology could be used for implementing the dynamic database. The distributed DBMS will handle concurrency and security control, and transparency in the access and update of information. Thus, applications do not need to take care of these tasks.

The main gain provided by this technology is that systems do not need to be in the same VAX cluster in order to share the database; databases can be located in different sites, and even in different hardware, and the distributed DBMS is responsible to get access to the appropriate database. Besides getting this new advantage, in this approach all the advantages listed under 9.2.1 still hold, namely reduction of message flow, and elimination of tape hands-off, duplicate maintenance, and duplicate processing. All this have an impact on the reduction of costs.

If this technology is used, two different approaches can be implemented, locate the daily dynamic database in the Transaction Processing component and the historic dynamic database in the Information Processing component; or the consolidation of the daily and historic database in IP. By following either of these approaches, the following additional advantages would be obtained.

9.3.1 PROBLEMS SOLVED BY USING DISTRIBUTED DBMSs

1. Reducing processing overhead.

By implementing distributed databases, a back-end processor can take care of all the database accesses. Therefore the front-end processors, which currently perform front-end and back-end tasks since they handle database accesses and updates (e.g. postings, query accesses, etc) would be further relieved of these tasks. As a consequence, front-end processors would have the capability of accomodating growth without having to add more processing power. For example, the Transaction Processor of the Funds Transfer System would be able to handle a larger volume of transactions without having to add more processing capacity. The extra capacity that would be obtained would be very useful since the demand for funds transfer transactions is increasing at a rate of 25% per year.

2. Share of data outside the VAX cluster.

With the distributed DBMS, applications can access the dynamic database even if they are not part of the same VAX cluster. This represents an alternative implementation to achieving the sharing of data. Currently, applications need to be in the same cluster in order to share a database, and there is a limit on the number of nodes that can share a VAX cluster. The current limit is 16 nodes per VAX cluster. Thus, by using distributed DBMS, data can be shared outside the VAX cluster

providing further flexibility in achieving integration of data.

3. Cash Management System.

The Cash Management system needs to access both, the daily dynamic database, and the historical database. The distributed DBMS takes care of the access of the required information, and neither the application nor the user needs to know where the data is actually located. Moreover, the Cash Management system could be located in either the TP or IP since there are no restrictions on location in order to share a database.

4. Demand Deposit Account System.

With the first approach, the daily dynamic database is located in the Transaction Processing cluster, and the historic dynamic database in the Information Processing component. This is illustrated in Figure 23. The DDA system would be able to access either of the dynamic databases transparently. However, the current distributed DBMS technology does not handle the updating of redundant databases (i.e., deferred copies), therefore the historic dynamic database would have to be updated daily with the postings that occurred in the daily database, and procedures need to be set up to handle the redundancy. When that technology becomes available, the historical DDA database can be automatically updated, - as frequently as needed, probably once a day, - with all the

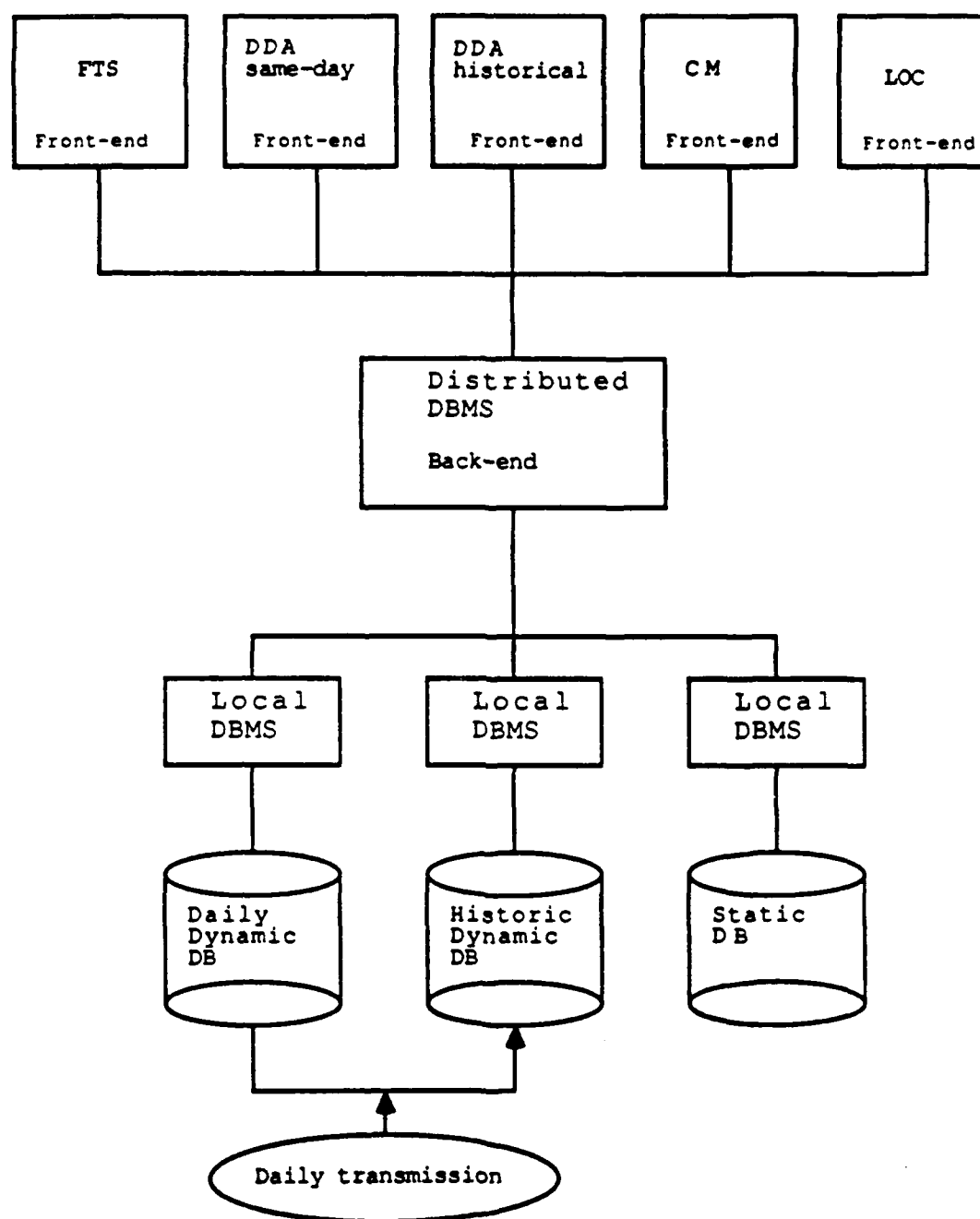


Figure 23

Distributed databases architecture

updates that have been done to the daily dynamic database. Thus, applications will not need to propagate the updates to other databases. Some distributed DBMS users believe that this feature should be available very soon.

With the second approach, the consolidation of dynamic databases in IP, the maintenance of the historical dynamic database, as a result of the changes occurring in the daily dynamic database would be completely eliminated since both databases are consolidated. Moreover, all the operational functions required to maintain both databases is eliminated. Distributed DBMS vendors argue that there is performance transparency, this means that performance does not depend on location. Thus, placing the dynamic database in IP should not degrade FTS performance (i.e. postings). However, the duplication of the daily database in FTS for performance reasons would only be possible when the commercial distributed DBMSs support deferred copies. In this case, the propagation of updates will be taken care by the distributed DBMS. However, if data is duplicated in different systems, issues such as, concurrency control, and locking, need to be taken care of in the redundant databases. Hopefully, the technology to become available supporting deferred copies will take care of these issues.

In sum, distributed databases seems to be an alternative to be considered for future directions. It reduces costs of

maintenance and operation by at least the same amount as the previous alternative. Moreover, if applications, such as the Investigations System is included in this architecture, investigation queries will reflect the books of the bank in real time.

Distributed databases enable the sharing of data without imposing restrictions on the location of the data. This removes the technological constraint imposed by the maximum limit of nodes supported by a VAX cluster. The main advantage of this approach is to provide flexibility in the sharing and access of databases by systems located in different sites. The main gains that would be obtained from it are elimination of duplicate processing, shadow postings, tape hands-off among systems, duplicate maintenance, and duplicate information which implies a waste of storage. All these will have an effect on the ability to reduce operational costs besides providing a much better environment to manage risk control and handle growth and evolution.

Conclusions

This thesis evaluates the validity of the conceptual model as the means to meet the goals of the Financial Institution organization. Although the conceptual model is a good theoretical vehicle that compromises the three CIS principles, namely, integration, autonomy, and evolution, its implementation has shown to be weak in some aspects, especially in those concerning integration.

Integration has been partially accomplished by building a communications network infrastructure that allows to communicate across independent systems. However, it has failed to provide an infrastrucure for the integration of data. As a result, independent databases have been created, some of which contain redundant data.

The redundancy of data brings several issues, consistency of data, propagation of updates, duplication of efforts, tape hands-off, duplication of processing, etc. All of these have an effect on the cost to maintaining these systems. Autonomy, on the other hand, has been accomplished by allowing these systems to be developed independently, most of them being hosted in their own hardware, both of these factors have resulted in their succesful implementation. However, autonomy has been mediated by the setting of hardware and software standards which will facilitate integration. Finally, these

systems are better prepared to accomodate evolution, mainly because there has been some level of systems aggregation, and some of the technical constraints imposed by the previous architecture have been removed.

The main question that remains opened and which needs to be addressed is the integration of the data. By reviewing the Composite Information System methodology, the main technical obstacles identified are, the dispersion and redundancy of the databases across the FI systems, and the lack of use of DBMS technology. The main organizational obstacles are, the high degree of local autonomy, the desire for immediate results, and budget constraints. These obstacles have an effect on the cost of operations, maintenance, and future performance of these systems. Integration was identified as a critical factor to reduce these costs in the long run. Even if integration of data may not be perceived as an immediate need, it may either jeopardize the ability of the systems to grow, or their maintenance can become a burden that may constraint the need to meet new requirements. Moreover, in order for these systems to accomodate growth without incurring future high costs of development, maintenance of the systems, and lost of operations due to a non competitive service, these systems should accomodate to the integration goal.

Integration can only be truly accomplished by consolidating the data, either physically or logically. The use of DBMS

technology would take care of the following issues, concurrency control, security, and transparency in the access of data. Thus, DBMS will alleviate the applications programs from performing these tasks which will have an effect on ease of development, and ease of maintenance. Although it has been argued that the high performance required by these systems does not permit the use of DBMS technology, in the particular case of the dynamic database, the use of DBMS technology may be feasible. Consolidating the dynamic database can also provide advantages for risk control, since checks can be performed against the current "bank books".

Another approach to be explored is that of distributed databases. This technology allows the sharing of segregated databases located in different sites by different applications. This technology removes the technical constraint of having to belong to the same VAX cluster in order to share a database. Moreover, the burden imposed by concurrency and security controls, and access to the data would be handled by the database manager. Although this technology does not allow yet for automatic updating of redundant databases, this technology is expected to emerge soon. When this technology is available, databases can be spread out allowing for redundancy to obtain higher performance -if needed-. Moreover, this databases are expected to increase in performance, and are currently available for the hardware architecture used in the bank,

namely VAXes and the MVS operating system.

Until that technology becomes more mature, other roads can be explored. The split of the dynamic database into daily database, and historical database seems to be the most plausible alternative. This requires moving the daily processing DDA, the LOC, and the CM systems to the Transaction Processor (TP) cluster. Moreover, the movement of the daily processing of DDA has already been considered in the Strategic Systems Plan for Financial Institution Systems. By moving DDA to TP along with LOC and CM, sharing of the daily dynamic database would be possible, eliminating all the costs incurred by having to update in real time the otherwise redundant dynamic databases. In this case, the TP component would have all the current (i.e., daily data) whereas the IP component would have all the DDA historic data. This is also consistent with the conceptual model. By using DBMS technology to implement the shared database, a further step towards the full integration of data is accomplished.

References

- [1] Madnick, Stuart and Wang, Y. Richard. "Gaining Strategic Advantage through Composite Information Systems." Sloan School of Management, working paper, 1987.
- [2] Madnick, Stuart and Wang, Y. Richard. "Strategic applications of Very large databases through Composite Information Systems." Sloan School of Management, working paper, 1987.
- [3] Benjamin, Robert; Rockart, John; Scott Morton, Michael; and Wyman, John. "Information Technology: A strategic opportunity." Sloan Management Review, Spring 84.
- [4] "Note on Airline Reservation Systems (Condensed)." Harvard Business School, 1984.
- [5] Rockart, Jack; Bullen, Christine; and Kogan, John. "The Management of Distributed Processing." CISR WP #39, December 1978.
- [6] Massimo, Joseph. "Gaining Strategic Advantage through Composite Information Systems. A case study." Sloan School of Management, Master's thesis, May 1987.
- [7] "Product Information." Internal bank's documents.
- [8] Frank, William; Madnick, Stuart and Wang, Y. Richard. "A conceptual model for integrated autonomous processing: An international bank's experience with large databases." Sloan School of Management, Working paper, March 1987.
- [9] "Funds Transfer Systems (FTS) documentation." Internal bank's documents. December, 1984.

- [10] "Demand Deposit Account (DDA) documentation." Internal bank's documents. March, 1985.
- [11] Porter, Michael E. "Competitive Advantage." The Free Press, 1985.

Bibliography

1. Akoka, Jacob. "Centralization vs. Decentralization of Information systems: A critical survey and an annotated bibliography." CISR WP #36, November 1977.
2. Benjamin, Robert and Scott Morton, Michael. "Information Technology, integration, and organizational change." CISR WP #138, April 1986.
3. Emery, James. "Integrated information systems and their effects on organizational structure." Grochla, E. and Szyperski, N. Information Systems and organizational structure, pp. 95-103, 1975.
4. Greff, Robert. "Distributed Database Systems: A comparison between ORACLE and INGRES." Sloan School of Management, Master's thesis, May 1987.
5. Keen, Peter; Bronsema, Gloria; and Zuboff, Shoshanna. "Implementing common systems: One organization's experience." CISR WP #80, August 1981.
6. McFarlan, F. Warren. "Information Technology changes the way you compete." Harvard Business Review, May-June 1984.
7. McKenney, James and McFarlan, F. Warren. "The information archipelago: maps and bridges." Harvard Business Review, September-October 1982.
8. Parsons, Gregory. "Information Technology: A new competitive weapon." Sloan Management Review, Fall 1983.

9. Weick, Karl E. "Educational organizations as loosely coupled systems." *Administrative Science Quarterly*, Vol. 21, March 1976.

END